DISJUNCTIVE NORMAL FORMULA BASED SUPERVISORY CONTROL POLICY FOR GENERAL PETRI NETS

BY

VIJAYALAKSHMI DEVERAKONDA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Associate Professor Ramavarapu S. Sreenivas

ABSTRACT

A *Petri net* (PN) is said to be *live* if it is possible to fire any transition, although not immediately, from every reachable marking. A *liveness enforcing supervisory policy* (LESP) determines which controllable transition is to be prevented from firing at a marking, to ensure the supervised *Petri net* (PN) is live.

A LESP is said to be *minimally restrictive* if the following property is true – if a minimally restrictive LESP prevents the firing of a transition at a marking, then all other LESPs should do the same.

We restrict our attention to a class of general *Petri nets* (PN) structures, where the existence of an LESP for an instance initialized at a marking, implies the existence of an LESP when the same instance is initialized with a larger initial marking. We show that the minimally restrictive LESP for an instance N from this class is characterized by a collection of boolean formulae $\{\Theta_{t_c}(N)\}_{t_c \in T_c}$, where T_c is the set of controllable transitions in the PN. The literals in $\Theta_{t_c}(N)$ are true if and only if the token-load of specific places meet a threshold. Consequently, appropriately placed *threshold-sensors*, which detect if the token-load of a place is greater than or equal to a predetermined threshold, provide sufficient information to implement the minimally restrictive LESP.

For my parents, my grandparents and my Adviser, Professor Ramavarupu S. Sreenivas

ACKNOWLEDGMENTS

I would like to first and foremost thank my Adviser, Professor Ramavarupu S. Sreenivas, without whose guidance, this work would not have been possible. I am also grateful to the Department of Industrial and Enterprise Systems Engineering for funding me during my Graduate studies.

I would also like to thank Praveen Tumuluri and Sweta Yamini Seethamaraju, friends whose support and insights were indispensable. Last but not least, I would also like to thank my parents and grandparents for their love and encouragement to pursue my Graduate studies.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A Petri net (PN) is a graphical tool used in the field of mathematical modeling of *Discrete-Event/Discrete-State* (DEDS) systems. PNs are used to represent parallel, concurrent, and/or stochastic systems. This tool resembles a flowchart and therefore used for visually describing such systems. There are two classes of nodes in a graphical description of a PN *places* and *transitions*. Transitions are used to represent activities that occur in the system. *Tokens*, which are usually represented by filled-circles that reside in places, represent the present state of the system. In addition, there are *arcs* that emanate either from a place to a transition, or from a transition to a place. We say a place is an *input* to a transition if there is an arc from the place to the transition. Similarly, we say a place is an *output* to a transition if there is an arc from the transition to the place. The set of places, transitions and arcs define the *structure* of the PN. The PN is fully defined when all places are initialized with tokens. That is, a PN structure, along with an initial distribution of tokens among the places, forms the PN. The term *marking* is used to denote the distribution of tokens among the places in a PN.

The activity represented by a transition can occur if there are a sufficient number of tokens in each of its input places. The *firing* of the associated transition represents the occurrence of this event. This instantaneous activity results in an appropriate number of tokens being taken from each input place, followed by the placement of a relevant number of tokens in each output place of the transition. That is, each firing of a transition changes the marking, or the distribution of tokens among the places. It is possible to describe the change in the marking using a state equation. Prof. Carl Adam Petri invented Petri nets in 1962, while he was working on his dissertation at the Technical University of Darmstadt, West Germany.

Due to the generality of Petri nets, they are used for a wide variety of applications that can be used to describe concurrent processes. They can be used to illustrate almost any system, including those systems that contain concurrent activities. A disadvantage of using Petri net based models is that the analysis of even simple systems can be complicated in some cases. Therefore, one must carefully assess whether Petri nets are suitable for the system currently

observed.

Some applications of Petri nets include flexible manufacturing/ Industrial control systems, concurrent programs, programmable logic and VLSI arrays, modeling and analysis of distributed software systems, digital filters, multi-processor memory systems, fault-tolerant systems, office information systems, compiler and operating systems, and formal language. Petri nets have also been used in legal systems, human factors and decision models.

A discrete event dynamic system is said to be *live* if every event in a given model can occur at any point of time in the future, not necessarily immediately, irrespective of the past events that have occurred. This condition of liveness is relaxed to different levels due to the analytical complexity of certain systems such as the operating system of a computer or a large multi-component logistic systems. The level of liveness are dead (L0-live) where a transition cannot be fired in any firing sequence, L1-live (potentially fireable) where a transition can be fired at least once in a firing sequence, L2-live where a transition can be fired a finite number of times, L3-live where a transition can be fired an infinite number of times in a firing sequence and L4-live where a transition is L1-live for every marking reachable from the initial marking [4]. We concern ourselves with L4-liveness in this thesis.

Liveness property of Petri nets is essential because it indicates the absence of deadlocks in the system. But it is important to note that deadlock-free systems are not necessarily live. If the liveness property is satisfied, every transition in the net has the ability to fire at some point during the firing sequence.

The structure of the PN determines the set of transitions that can be fired at a given marking. Also, a supervisory policy determines which of these transitions can be permitted at any marking. A non-live system, described by a Petri net can be made live by supervision. There is a noteworthy amount of work in the literature that describes this process in detail. The process of synthesizing the liveness enforcing supervisory policy (LESP) for a PN is unsolvable in general. Consequently, attention has to be restricted to subclasses of PNs if this has to be done automatically/algorithmically.

A set of integral vectors is *right-closed* if the presence of a vector in the set would imply all vectors that are larger than it are also in the set. A right-closed set can be described by its minimal elements. The class of PNs considered in this thesis are characterized by the property that the set of initial markings for which there is an LESP is right-closed. The minimal elements of this right-closed set can be computed using existing software.

Prior work done with supervisory policies includes an object-oriented implementation of an algorithm that determines a *minimally restrictive* LESP [3, 1]. If the minimally restrictive

LESP prevents the firing of the transition at a given marking, then all of the other LESPs should also prevent the firing of the transition. The supervisory policy that prevents the firing of a transitions at a marking if the new marking that results is not in the right-closed set mentioned above, is the minimally restrictive LESP. This minimally restrictive LESP is denoted as $\mathcal{P}_2$ in this thesis.

The naïve implementation of $\mathcal{P}_2$ will require the comparison of a new marking with each minimal element before an appropriate control-action can be prescribed. This may be unnecessary in some cases. This task has to be performed efficiently, and this thesis has results that provide the required guidance on this matter. The first result is that it is possible to determine when a controllable transition is permanently enabled by the minimally restrictive LESP. For those controllable transitions that have to be disabled under some marking, there is a Disjunctive Normal Formula (DNF) whose truth-value depends on the present marking, which determines if the controllable transition is to be permitted at a marking. The literals in the DNF whose truth-value depends on the present marking, which determines if the controllable transition is to be permitted at a marking. The literals in the DNF identify whether the token loads of the places are greater than or equal to a certain threshold. This DNF is computed for each controllable transition, and its truth-value depends on the current marking. This computation is dependent on the right closed set mentioned earlier. A controllable transition is permitted at a marking if and only if the DNF is true at that marking. The policy determined from this DNF expression is referred to as $\mathcal{P}_1$ throughout the thesis. One of the main results of this thesis is that the policy synthesized from the DNF expression $\mathcal{P}_1$ and minimally restrictive LESP $\mathcal{P}_2$ are identical.

A threshold sensor is used to monitor the number of tokens at various places in the Petri net structure. As noted above, the DNF has literals that are thresholds which means one can identify a number of threshold-sensors, that when placed on appropriate places, is sufficient to enforce the DNF-based policy. This work has been conditionally accepted for publication in the IEEE Transactions on Automatic Control [5].

The rest of this thesis is organized as follows. Chapter 2 introduces the notations and definitions used in the remainder of this thesis. Section 2.3 presents the main results, which is followed by a collection of examples in chapter 3. The conclusions and future research directions are presented in chapter 4. The relevant code samples can be found in the appendix, which is chapter 6.

CHAPTER 2

NOTATIONS AND DEFINITIONS, SOME PRELIMINARY OBSERVATIONS AND MAIN RESULTS

We use $\mathcal{N}$ ($\mathcal{N}^+$) to denote the set of non-negative (positive) integers. The term $card(\bullet)$ denotes the cardinality of the set argument. A *Petri net structure* $N = (\Pi, T, \Phi, \Gamma)$ is an ordered 4-tuple, where $\Pi = \{p_1, \ldots, p_n\}$ is a set of n *places*, $T = \{t_1, \ldots, t_m\}$ is a collection of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *arcs*, and $\Gamma : \Phi \to \mathcal{N}^+$ is the *weight* associated with each arc. The weight of an arc is represented by an integer that is placed along side the arc. For brevity, we refrain from denoting the weight of those arcs $\phi \in \Phi$ where $\Gamma(\phi) = 1$. A PN structure is said to be *ordinary* (*general*) if the weight associated with an arc is (not necessarily) unitary.

The *initial marking function* (or the *initial marking*) of a PN structure N is a function $\mathbf{m}^0 : \Pi \to \mathcal{N}$, which identifies the number of *tokens* in each place. We will use the term *Petri net* (PN) and the symbol $N(\mathbf{m}^0)$ to denote a PN structure N along with its initial marking $\mathbf{m}^0$.

A marking $\mathbf{m} : \Pi \to \mathcal{N}$ is sometimes represented by an integer-valued vector $\mathbf{m} \in \mathcal{N}^n$, where the i-th component $\mathbf{m}_i$ represents the token-load ($\mathbf{m}(p_i)$) of the i-th place.

We define the sets ${}^\bullet x := \{y \mid (y, x) \in \Phi\}$ and $x^\bullet := \{y \mid (x, y) \in \Phi\}$. A transition $t \in T$ is said to be *enabled* at a marking $\mathbf{m}^i$ if $\forall p \in {}^\bullet t, \mathbf{m}^i(p) \geq \Gamma((p, t))$. The set of enabled transitions at marking $\mathbf{m}^i$ is denoted by the symbol $T_e(N, \mathbf{m}^i)$. An enabled transition $t \in T_e(N, \mathbf{m}^i)$ can *fire*, which changes the marking $\mathbf{m}^i$ to $\mathbf{m}^{i+1}$ according to $\mathbf{m}^{i+1}(p) = \mathbf{m}^i(p) - \Gamma(p, t) + \Gamma(t, p)$.

When the marking is interpreted as a nonnegative integer-valued vector, it is useful to define the *input matrix* $\mathbf{IN}$ (*output matrix* $\mathbf{OUT}$) as an $n \times m$ matrix, where $\mathbf{IN}_{i,j}$ ($\mathbf{OUT}_{i,j}$) equals $\Gamma((p_i, t_j))$ ($\Gamma((p_i, t_j))$) if $p_i \in {}^\bullet t_j$, ($p_i \in t_j^\bullet$) and is zero-value otherwise. The *incidence matrix* $\mathbf{C}$ of the PN N is an $n \times m$ matrix, where $\mathbf{C} = \mathbf{OUT} - \mathbf{IN}$.

A set of markings $\mathcal{M} \subseteq \mathcal{N}^n$ is said to be *right-closed* [6] if $((\mathbf{m}^1 \in \mathcal{M}) \wedge (\mathbf{m}^2 \geq \mathbf{m}^1) \Rightarrow (\mathbf{m}^2 \in \mathcal{M}))$. The set $\mathcal{M} \subseteq \mathcal{N}^n$ contains a finite set of minimal-elements, $min(\mathcal{M}) \subset \mathcal{M}$.

2.1 Supervisory Control of PNs

The paradigm of supervisory control of PNs assumes a subset of *controllable transitions*, denoted by $T_c \subseteq T$, which can be prevented from firing by an external agent called the *supervisor*. The set of *uncontrollable transitions*, denoted by $T_u \subseteq T$, is given by $T_u = T - T_c$. The controllable (uncontrollable) transitions are represented as filled (unfilled) boxes in graphical representation of PNs.

A *supervisory policy* $\mathcal{P} : \mathcal{N}^n \times T \rightarrow \{0, 1\}$, is a function that returns a 0 or 1 for each transition and each reachable marking. The supervisory policy $\mathcal{P}$ permits the firing of transition t_j at marking $\mathbf{m}^i$, only if $\mathcal{P}(\mathbf{m}^i, t_j) = 1$. If $t_j \in T_e(N, \mathbf{m}^i)$ for some marking $\mathbf{m}^i$, we say the transition t_j is *state-enabled* at $\mathbf{m}^i$. If $\mathcal{P}(\mathbf{m}^i, t_j) = 1$, we say the transition t_j is *control-enabled* at $\mathbf{m}^i$. A transition has to be state- and control-enabled before it can fire. The fact that uncontrollable transitions cannot be prevented from firing by the supervisory policy is captured by the requirement that $\forall \mathbf{m}^i \in \mathcal{N}^n, \mathcal{P}(\mathbf{m}^i, t_j) = 1$, if $t_j \in T_u$. This is implicitly assumed of any supervisory policy in this paper.

A string of transitions $\sigma = t_1 t_2 \cdots t_k$, where $t_j \in T (j \in \{1, 2, \ldots, k\})$ is said to be a *valid firing string* starting from the marking $\mathbf{m}^i$, if, (1) $t_1 \in T_e(N, \mathbf{m}^i), \mathcal{P}(\mathbf{m}^i, t_1) = 1$, and (2) for $j \in \{1, 2, \ldots, k-1\}$ the firing of the transition t_j produces a marking $\mathbf{m}^{i+j}$ and $t_{j+1} \in T_e(N, \mathbf{m}^{i+j})$ and $\mathcal{P}(\mathbf{m}^{i+j}, t_{j+1}) = 1$.

The set of reachable markings under the supervision of $\mathcal{P}$ in N from the initial marking $\mathbf{m}^0$ is denoted by $\Re(N, \mathbf{m}^0, \mathcal{P})$. If $\mathbf{m}^{i+k}$ results from the firing of $\sigma \in T^*$ starting from the initial marking $\mathbf{m}^i$, we represent it symbolically as $\mathbf{m}^i \xrightarrow{\sigma} \mathbf{m}^{i+k}$. If $\mathbf{x}(\sigma)$ is an m-dimensional vector whose i-th component corresponds to the number of occurrences of t_i in a valid string $\sigma \in T^*$, and if $\mathbf{m}^i \xrightarrow{\sigma} \mathbf{m}^{i+j}$, then $\mathbf{m}^{i+j} = \mathbf{m}^i + \mathbf{C}\mathbf{x}(\sigma)$.

A transition t_k is *live* under the supervision of $\mathcal{P}$ if

$$\forall \mathbf{m}^i \in \Re(N, \mathbf{m}^0, \mathcal{P}), \exists \mathbf{m}^j \in \Re(N, \mathbf{m}^i, \mathcal{P}) \text{ such that } t_k \in T_e(N, \mathbf{m}^j) \text{ and } \mathcal{P}(\mathbf{m}^j, t_k) = 1.$$

A policy $\mathcal{P}$ is a *liveness enforcing supervisory policy* (LESP) for $N(\mathbf{m}^0)$ if all transitions in $N(\mathbf{m}^0)$ are live under $\mathcal{P}$. The policy $\mathcal{P}$ is said to be *minimally restrictive* if for every LESP $\widehat{\mathcal{P}} : \mathcal{N}^n \times T \rightarrow \{0, 1\}$ for $N(\mathbf{m}^0)$, the following condition holds $\forall \mathbf{m}^i \in \mathcal{N}^n, \forall t \in T, \mathcal{P}(\mathbf{m}^i, t) \geq \widehat{\mathcal{P}}(\mathbf{m}^i, t)$.

For an arbitrary PN structure $N = (\Pi, T, \Phi, \Gamma)$, the set

$$\Delta(N) = \{\mathbf{m}^0 \in \mathcal{N}^{card(\Pi)} \mid \exists \text{ an LESP for } N(\mathbf{m}^0)\}$$

denotes the set of initial markings $\mathbf{m}^0$ for which there is a LESP for $N(\mathbf{m}^0)$. $\Delta(N)$ is *control invariant* (cf. proposition 7.1, [7]) with respect to N; that is, if $\mathbf{m}^1 \in \Delta(N), t_u \in T_e(N, \mathbf{m}^1) \cap T_u$ and $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^2$ in N, then $\mathbf{m}^2 \in \Delta(N)$. Equivalently, only the firing of a controllable transition at any marking in $\Delta(N)$ can result in a new marking that is not in $\Delta(N)$.

There is an LESP for $N(\mathbf{m}^0)$ if and only if $\mathbf{m}^0 \in \Delta(N)$. If $\mathbf{m}^0 \in \Delta(N)$, the LESP that prevents the firing of a controllable transition at any marking when its firing would result in a new marking that is not in $\Delta(N)$, is the minimally restrictive LESP for $N(\mathbf{m}^0)$ (cf. Lemma 5.9, [8]). We use the symbol $\mathcal{P}_N^*$ to denote this minimaly restrictive LESP. The existence of an LESP for an arbitrary PN is undecidable (cf. corollary 5.2, [9]), and is decidable if all transitions in the PN are controllable, or if the PN structure N belongs to the classes identified in the following references [8, 10, 11]. The process of deciding the existence of an LESP in an arbitrary instance from these classes is NP-hard.

2.2 Review of Relevant Prior Work

Monitors are places added to an existing PN structure, whose token-load at any instant indicates the amount of a particular resource that is available for consumption. The input and output arcs to this place appropriately capture the consumption and production of resources in the original PN. These were originally introduced into supervisory control of PNs by Giua [12] to handle mutual exclusion constraints. Moody and Antsaklis represent liveness constraints in specific PNs as linear inequalities, which are then implemented using monitor places. This work was extended by Iordache and Antsaklis to include a sufficient condition for the existence of policies that enforce liveness in a class of PNs called *Asymmetric Choice Petri nets*[1] [13].

Structural features of a PN, known as *siphons*, characterize the liveness of some classes of PNs. Several authors have used monitor place constructions that prevent siphons from being undermarked (cf. [14, 15], for example). References [16, 17] uses a set of inequalities to characterize insufficiently marked siphons that is subsequently used to develop an algebraic LESP-synthesis procedure. Li et al [18] develop an iterative siphon-based control scheme for preventing deadlocks in PN models of manufacturing systems using a mixed integer programming approach involving what are known as *necessary siphons*.

[1] cf. page 554, [4] for a formal definition.

Reveliotis [19] developed a class of policies for resource allocation systems that can be extended to the PN-framework using the *theory of regions*. Ghaffari, Rezg and Xie [20] also use the theory of regions to obtain a minimally restrictive supervisory policy that enforces liveness for a class of PNs. Marchetti and Munier-Kordon [21] presented a sufficient condition for liveness, that can be tested in polynomial time, for a class of general PNs known as *Unitary Weighted Event Graphs*. Basile et al. [22] presented sufficient conditions for minimally-restrictive, closed-loop liveness of a class of *Marked Graph* PNs supervised by monitors that enforce *Generalized Mutual Exclusion Constraints* (GMECs).

We present the main results of this paper in the next section.

2.3 Main Results

As noted earlier, for any PN $N(\mathbf{m}^0)$, where $\mathbf{m}^0 \in \Delta(N)$, the minimally restrictive LESP $\mathcal{P}_N^*$ disables a controllable transition $t_c \in T_c$ at a marking $\mathbf{m}^1 \in \Re(N, \mathbf{m}^0, \mathcal{P}_N^*)$ if and only if $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$ in $N(\mathbf{m}^1)$ and $\mathbf{m}^2 \notin \Delta(N)$.

For the remainder of this paper, we restrict attention to PN structures N, where $\Delta(N)$ is right-closed. The finite set of minimal elements of $\Delta(N)$, $\min(\Delta(N))$, can be computed for this class of PN structures (cf. [3, 2, 1]). The remainder of this paper is about characterizing the minimally restrictive LESP $\mathcal{P}_N^*$ using boolean expressions that involve inequalities on the token-loads of places, which in turn yields a sufficient sensing strategy for the implementation of the minimally restrictive LESP.

The following observation identifies the controllable transitions that are never control-disabled by the minimally restrictive LESP $\mathcal{P}_N^*$.

Lemma 2.3.1. *Let $N = (\Pi, T, \Phi, \Gamma)$ be a PN structure where $\Delta(N)$ is right-closed. Suppose $\min(\Delta(N)) = \{\widehat{\mathbf{m}}_1, \widehat{\mathbf{m}}_2, \ldots, \widehat{\mathbf{m}}_k\}$, where each $\widehat{\mathbf{m}}_i \in \mathcal{N}^n$ and $card(\Pi) = n$. A controllable transition $t_c \in T_c$ is control-enabled by the minimally restrictive LESP $\mathcal{P}_N^*$ at any marking in $\Delta(N)$ if and only if $\forall \widehat{\mathbf{m}}_i \in \min(\Delta(N))$, $\exists \widehat{\mathbf{m}}_j \in \min(\Delta(N))$, such that*

$$\max\{\mathbf{IN}_{\bullet,c}, \widehat{\mathbf{m}}_i\} + \mathbf{C} \times \mathbf{1}_c \geq \widehat{\mathbf{m}}_j,$$

where $\mathbf{IN}_{\bullet,c}$ denotes the c-th column of the input matrix $\mathbf{IN}$, and $\mathbf{1}_c$ is the unit-vector where the c-th element is unity.

Proof. (*If*) If for some $\widehat{\mathbf{m}}_i \in \min(\Delta(N))$, $\exists \widehat{\mathbf{m}}_j \in \min(\Delta(N))$, such that

$$\max\{\mathbf{IN}_{\bullet,c}, \widehat{\mathbf{m}}_i\} + \mathbf{C} \times \mathbf{1}_c \geq \widehat{\mathbf{m}}_j.$$

If $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$ and $\mathbf{m}^1 \geq \widehat{\mathbf{m}}_i (\Rightarrow \mathbf{m}^1 \in \Delta(N))$, then $\mathbf{m}^2 \in \Delta(N)$. Consequently, $\mathcal{P}_N^*$ will not control-disable t_c at the marking $\mathbf{m}^1$. The result follows from the fact that that above claim holds $\forall \widehat{\mathbf{m}}_i \in \min(\Delta(N))$.

(*Only If*) If $\exists \widehat{\mathbf{m}}_i \in \min(\Delta(N))$, $\forall \widehat{\mathbf{m}}_j \in \min(\Delta(N))$, such that

$$\max\{\mathbf{IN}(\bullet, c), \widehat{\mathbf{m}}_i\} + \mathbf{C} \times \mathbf{1}_c \ngeq \widehat{\mathbf{m}}_j.$$

Then, for marking $\mathbf{m}^1 = \max\{\mathbf{IN}(\bullet, c), \widehat{\mathbf{m}}_i\} (\Rightarrow \mathbf{m}^1 \in \Delta(N))$, the minimally restrictive LESP $\mathcal{P}_N^*$ will control disable transition $t_c \in T_c$ at marking $\mathbf{m}^1 \in \Delta(N)$. □

To illustrate lemma 2.3.1, we consider the FCPN N_1 structure shown in figure 2.1(a). We have $\widehat{\mathbf{m}}_1 = (1\ 0\ 0\ 0\ 0)^T$ and $\widehat{\mathbf{m}}_2 = (0\ 0\ 0\ 1\ 1)^T$. For transition $t_3 \in T_c$, we note that the conditions of lemma 2.3.1 is satisfied as

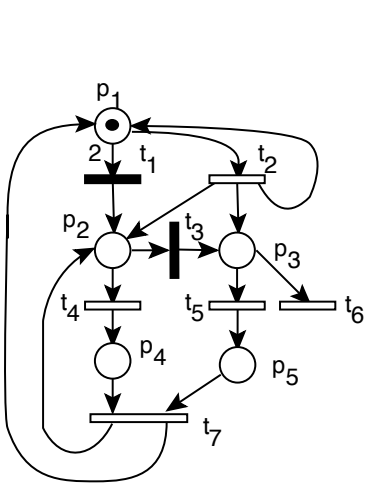
$$\max\left\{\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix}\right\} + \begin{pmatrix}0\\-1\\1\\0\\0\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\0\end{pmatrix} \geq \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix}, \text{and}$$

$$\max\left\{\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\0\\0\\1\\1\end{pmatrix}\right\} + \begin{pmatrix}0\\-1\\1\\0\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\1\\1\end{pmatrix} \geq \begin{pmatrix}0\\0\\0\\1\\1\end{pmatrix}.$$

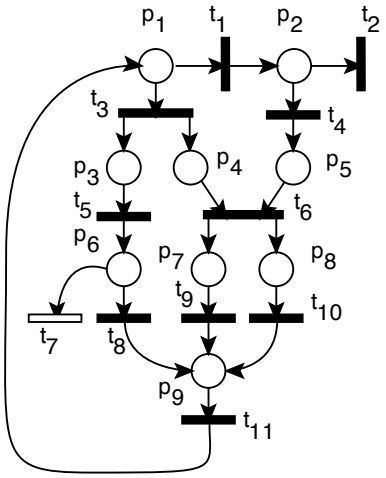
Consequently, transition t_3 is never control-disabled by the minimally restrictive LESP for $N_1(\mathbf{m}_1^0)$.

As an additional illustration, the controllable transitions $t_4, t_5, t_6, t_8, t_9, t_{10}$ and t_{11} in the PN structure N_2 shown in figure 2.1(b), meet the requirement of lemma 2.3.1 (cf. figure 2.1(c), which lists the members of $\min(\Delta(N_2))$. Consequently, these controllable transitions are never control-disabled by a minimally restrictive LESP for $N_2(\mathbf{m}_2^0)$.

On the flip-side, the controllable transition t_1 of N_1 does not meet the requirements of



(a) General FCPN structure N_1



(b) FCPN structure N_2

```
 1: ( 0 0 0 1 1 0 0 0 0 )
 2: ( 0 1 0 1 0 0 0 0 0 )
 3: ( 2 0 0 0 0 0 0 0 0 )
 4: ( 1 1 0 0 0 0 0 0 0 )
 5: ( 1 0 0 1 0 0 0 0 0 )
 6: ( 1 0 0 0 1 0 0 0 0 )
 7: ( 1 0 0 0 0 0 1 0 0 )
 8: ( 1 0 0 0 0 0 0 1 0 )
 9: ( 1 0 0 0 0 0 0 0 1 )
10: ( 0 1 0 0 0 0 1 0 0 )
11: ( 0 0 0 1 0 0 1 0 0 )
12: ( 0 0 0 0 1 0 1 0 0 )
13: ( 0 0 0 0 0 0 2 0 0 )
14: ( 0 0 0 0 0 0 1 1 0 )
15: ( 0 0 0 0 0 0 1 0 1 )
16: ( 0 1 0 0 0 0 0 1 0 )
17: ( 0 0 0 1 0 0 0 1 0 )
18: ( 0 0 0 0 1 0 0 1 0 )
19: ( 0 0 0 0 0 0 0 2 0 )
20: ( 0 0 0 0 0 0 0 1 1 )
21: ( 0 1 0 0 0 0 0 0 1 )
22: ( 0 0 0 1 0 0 0 0 1 )
23: ( 0 0 0 0 1 0 0 0 1 )
24: ( 0 0 0 0 0 0 0 0 2 )
```

(c) $\min(\Delta(N_2))$

Figure 2.1: (a) A General FCPN structure N_1, where $min(\Delta(N_1)) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 1\ 1)^T\}$. (b) An FCPN structure N_2, and (c) The twenty-four elements of $\min(\Delta(N_2))$ generated by the software described in references [1, 2, 3].

lemma 2.3.1. It is control-disabled by the minimally restrictive LESP for $N_1(\mathbf{m}_1^0)$ for $\mathbf{m}_1^0 = (2\ 0\ 0\ 0\ 0)^T (\in \Delta(N_1))$. Likewise, the controllable transitions t_1 and t_2 will be control-disabled by the minimally restrictive LESP for $N_2(\mathbf{m}_2^0)$ for $\mathbf{m}_2^0 = (1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T (\in \Delta(N_2))$; t_3 is control-disabled by the minimally restrictive LESP for $N_2(\mathbf{m}_2^0)$ for $\mathbf{m}_2^0 = (1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)^T (\in \Delta(N_2))$.

We turn our attention to those controllable transitions $t_c \in T_c$, that do not satisfy the requirement of lemma 2.3.1, and would have to be control-disabled at some marking in $\Delta(N)$. Using the procedure of figure 2.2, we associate a *disjunctive-normal-form* (DNF) expression, $\Theta_{t_c}(N)$, with at most k *clauses*, where the *literals* ("$\mathbf{m}(p_i) \geq \beta$") identify if the token-load of a place (p_i) is greater than or equal to a threshold (β). In this context, $\Theta_{t_c}(N)$ can be viewed as a mapping $\Theta_{t_c}(N) : \mathcal{N}^n \rightarrow \{0, 1\}$, that returns a logical value for each marking. The controllable transition t_c is permitted at marking $\mathbf{m}$ if and only if $\Theta_{t_c}(N)(\mathbf{m}) = 1$. The main result of the paper shows that the control effected using $\Theta_{t_c}(N)$ for each $t_c \in T_c$ that violates the condition of lemma 2.3.1, is equivalent to the minimally restrictive LESP $\mathcal{P}_N^*$.

To motivate the main result, we note that when the procedure of figure 2.2 is executed on the PN structure N_1 of figure 2.1(a) and the controllable transition t_1, we obtain the

(DNF expression) *Compute-DNF* ((PN Structure) N, (controllable transition) $t_c \in T_c$)

Compute $\min(\Delta(N)) = \{\widehat{\mathbf{m}}_1, \widehat{\mathbf{m}}_2, \ldots, \widehat{\mathbf{m}}_k\}$
$\Theta_{t_c}(N) = \emptyset$
for $i = \{1, 2, \ldots, k\}$ **do**
    $Clause_i = \emptyset$ {/* Current Clause */}
    for $j = \{1, 2, \ldots, n\}$ **do**
        $\beta = \max\{0, (\widehat{\mathbf{m}}_i(p_j) + \mathbf{IN}_{j,c} - \mathbf{OUT}_{j,c})\}$
        if $((\beta > \mathbf{IN}_{j,c})$ && $(Clause_i == \emptyset))$ **then**
            $Clause_i = (\mathbf{m}(p_j) \geq \beta)$
        end if
        if $((\beta > \mathbf{IN}_{j,c})$ && $(Clause_i != \emptyset))$ **then**
            $Clause_i \leftarrow \Theta_{t_c}(N) \wedge (\mathbf{m}(p_j) \geq \beta)$
        end if
    end for
    if $((Clause_i != \emptyset)$ && $(\Theta_{t_c}(N) == \emptyset))$ **then**
        $\Theta_{t_c}(N) \leftarrow (Clause_i)$
    end if
    if $((Clause_i != \emptyset)$ && $(\Theta_{t_c}(N) != \emptyset))$ **then**
        $\Theta_{t_c}(N) \leftarrow \Theta_{t_c}(N) \vee (Clause_i)$
    end if
end for
Return $\Theta_{t_c}(N)$

Figure 2.2: The function *compute-DNF* (N, t_c) returns a DNF formula, $\Theta_{t_c}(N)$, with at most k-many clauses, where the literals take the form $(\mathbf{m}(p_i) \geq \beta)$, where $\beta \in \mathcal{N}^+$.

following DNF expression

$$\Theta_{t_1}(N_1) = (\mathbf{m}(p_1) \geq 3) \vee ((\mathbf{m}(p_4) \geq 1) \wedge (\mathbf{m}(p_5) \geq 1)).$$

Suppose $\mathbf{m}^1$ is a marking such that $t_1 \in T_e(N_1, \mathbf{m}^1)$ and either $(\mathbf{m}^1(p_1) \geq 3)$ or $((\mathbf{m}^1(p_4) \geq 1) \wedge (\mathbf{m}^1(p_5) \geq 1))$, then if $\mathbf{m}^1 \stackrel{t_1}{\rightarrow} \mathbf{m}^2$ in N_1, $\mathbf{m}^2 \in \Delta(N_1)$. That is, the controllable transition t_1 will be control-enabled under the minimally restrictive LESP $\mathcal{P}^*_{N_1}$ at the marking $\mathbf{m}^1$ also. If $\mathbf{m}^1$ is a marking such that $t_1 \in T_e(N_1, \mathbf{m}^1)$ and $(\mathbf{m}^1(p_1) \leq 2) \wedge ((\mathbf{m}^1(p_4) = 0) \vee (\mathbf{m}^1(p_5) = 0))$, then the minimally restrictive LESP $\mathcal{P}^*_{N_1}$ will not permit the firing of t_1 at marking $\mathbf{m}^1$. Therefore, the policy that permits t_1 only when the DNF $\Theta_{t_1}(N_1)$ is true is equivalent to the minimally restrictive LESP $\mathcal{P}^*_{N_1}$.

A *β-threshold sensor* at place $p \in \Pi$ produces an output of unity at a marking $\mathbf{m}$ if $\mathbf{m}(p) \geq \beta$; it produces a output of zero otherwise. The structure of the DNF $\Theta_{t_1}(N_1)$, together with the above observation, indicates that 3-threshold sensor at p_1, and 1-threshold sensors at p_4 and p_5 respectively, provides sufficient information for minimally restrictive supervision for liveness in $N_1(\mathbf{m}^0_1)$ for any $\mathbf{m}^0_1 \in \Delta(N_1)$.

```
( (m(p4)>=1) && (m(p5)>=1) ) ||
( (m(p4)>=1) ) ||
( (m(p1)>=3) ) ||
( (m(p1)>=2) ) ||
( (m(p1)>=2) && (m(p4)>=1) ) ||
( (m(p1)>=2) && (m(p5)>=1) ) ||
( (m(p1)>=2) && (m(p7)>=1) ) ||
( (m(p1)>=2) && (m(p8)>=1) ) ||
( (m(p1)>=2) && (m(p9)>=1) ) ||
( (m(p7)>=1) ) ||
( (m(p4)>=1) && (m(p7)>=1) ) ||
( (m(p5)>=1) && (m(p7)>=1) ) ||
( (m(p7)>=2) ) ||
( (m(p7)>=1) && (m(p8)>=1) ) ||
( (m(p7)>=1) && (m(p9)>=1) ) ||
( (m(p8)>=1) ) ||
( (m(p4)>=1) && (m(p8)>=1) ) ||
( (m(p5)>=1) && (m(p8)>=1) ) ||
( (m(p8)>=2) ) ||
( (m(p8)>=1) && (m(p9)>=1) ) ||
( (m(p9)>=1) ) ||
( (m(p4)>=1) && (m(p9)>=1) ) ||
( (m(p5)>=1) && (m(p9)>=1) ) ||
( (m(p9)>=2) )
```

(a) $\Theta_{t_1}(N_2)$

```
( (m(p4)>=1) && (m(p5)>=1) ) ||
( (m(p2)>=2) && (m(p4)>=1) ) ||
( (m(p1)>=2) ) ||
( (m(p1)>=1) && (m(p2)>=2) ) ||
( (m(p1)>=1) && (m(p4)>=1) ) ||
( (m(p1)>=1) && (m(p5)>=1) ) ||
( (m(p1)>=1) && (m(p7)>=1) ) ||
( (m(p1)>=1) && (m(p8)>=1) ) ||
( (m(p1)>=1) && (m(p9)>=1) ) ||
( (m(p2)>=2) && (m(p7)>=1) ) ||
( (m(p4)>=1) && (m(p7)>=1) ) ||
( (m(p5)>=1) && (m(p7)>=1) ) ||
( (m(p7)>=2) ) ||
( (m(p7)>=1) && (m(p8)>=1) ) ||
( (m(p7)>=1) && (m(p9)>=1) ) ||
( (m(p2)>=2) && (m(p8)>=1) ) ||
( (m(p4)>=1) && (m(p8)>=1) ) ||
( (m(p5)>=1) && (m(p8)>=1) ) ||
( (m(p8)>=2) ) ||
( (m(p8)>=1) && (m(p9)>=1) ) ||
( (m(p2)>=2) && (m(p9)>=1) ) ||
( (m(p4)>=1) && (m(p9)>=1) ) ||
( (m(p5)>=1) && (m(p9)>=1) ) ||
( (m(p9)>=2) )
```

(b) $\Theta_{t_2}(N_2)$

```
( (m(p5)>=1) ) ||
( (m(p2)>=1) ) ||
( (m(p1)>=3) ) ||
( (m(p1)>=2) && (m(p2)>=1) ) ||
( (m(p1)>=2) ) ||
( (m(p1)>=2) && (m(p5)>=1) ) ||
( (m(p1)>=2) && (m(p7)>=1) ) ||
( (m(p1)>=2) && (m(p8)>=1) ) ||
( (m(p1)>=2) && (m(p9)>=1) ) ||
( (m(p2)>=1) && (m(p7)>=1) ) ||
( (m(p7)>=1) ) ||
( (m(p5)>=1) && (m(p7)>=1) ) ||
( (m(p7)>=2) ) ||
( (m(p7)>=1) && (m(p8)>=1) ) ||
( (m(p7)>=1) && (m(p9)>=1) ) ||
( (m(p2)>=1) && (m(p8)>=1) ) ||
( (m(p8)>=1) ) ||
( (m(p5)>=1) && (m(p8)>=1) ) ||
( (m(p8)>=2) ) ||
( (m(p8)>=1) && (m(p9)>=1) ) ||
( (m(p2)>=1) && (m(p9)>=1) ) ||
( (m(p9)>=1) ) ||
( (m(p5)>=1) && (m(p9)>=1) ) ||
( (m(p9)>=2) )
```

(c) $\Theta_{t_3}(N_2)$

Figure 2.3: (a) The DNF-expression $\Theta_{t_1}(N_2)$ that is obtained when the procedure in figure 2.2 is executed on the PN structure N_2 (cf. figure 2.1(b)) and transition t_1. (b) The DNF-expression $\Theta_{t_2}(N_2)$, and (c) The DNF-expression $\Theta_{t_3}(N_2)$.

The DNF-expression $\Theta_{t_1}(N_2)$ that results when the procedure of figure 2.2 is executed

on the PN structure N_2 of figure 2.1(b) for the controllable transition t_1 is shown in figure 2.3(a). Each of the twenty-four clauses in the DNF-expression $\Theta_{t_1}(N_2)$ correspond to a member of $\min(\Delta(N_2))$, shown in figure 2.1(c). It can be seen that if the i-th clause is satisfied by a marking $\mathbf{m}^1$ where $t_1 \in T_e(N_2, \mathbf{m}^1)$, and $\mathbf{m}^1 \xrightarrow{t_1} \mathbf{m}^2$, then $\mathbf{m}^2$ is greater than or equal to the i-th member of $\min(\Delta(N_2))$. Consequently, if $\Theta_{t_1}(N_2)$ is true for a marking $\mathbf{m}^1 \in \Delta(N_2)$, transition t_1 will be control-enabled at the marking $\mathbf{m}^1$ by the minimally restrictive LESP $\mathcal{P}^*_{N_2}$. Conversely, if t_1 is control-enabled by $\mathcal{P}^*_{N_2}$ at a marking $\mathbf{m}^1 \in \Delta(N_2)$ and $\mathbf{m}^1 \xrightarrow{t_1} \mathbf{m}^2$, then by the procedure of figure 2.1, $\exists i \in \{1, 2, \ldots, k\}$ such that $\mathbf{m}^2 \geq \widehat{\mathbf{m}}_i$, and $\Theta_{t_1}(N_2)$ is true for $\mathbf{m}^1$. Similar observations can be made regarding the DNF-expressions $\Theta_{t_2}(N_2)$ and $\Theta_{t_3}(N_2)$ shown in figure 2.3(b) and 2.3(c), respectively. The supervisory policy of control-enabling transitions t_1, t_2 and t_3 in N_2 at a marking $\mathbf{m}^1 \in \Delta(N)$ if and only if the DNF-expressions for $\Theta_{t_1}(N_2), \Theta_{t_2}(N_2)$ and $\Theta_{t_3}(N_2)$ are true at marking $\mathbf{m}^1$ is equivalent to the minimally restrictive LESP $\mathcal{P}^*_{N_2}$. This observation is generalized as the main result of this paper, and the following observation is used in its proof.

Observation 2.3.2. *($Clause_i == \emptyset$) at line 14 of the procedure of figure 2.2 for the computation of $\Theta_{t_c}(N)$ if and only if $\mathbf{OUT}_{\bullet,c} \geq \widehat{\mathbf{m}}_i$.*

Proof. *(If)* If $\mathbf{OUT}_{\bullet,c} \geq \widehat{\mathbf{m}}_i$, then $\widehat{\mathbf{m}}_i + \mathbf{IN}_{\bullet,c} - \mathbf{OUT}_{\bullet,c} \leq \mathbf{IN}_{\bullet,c}$. Consequently, for each $j \in \{1, 2, \ldots, n\}$, at line 6 of the procedure of figure 2.2, $\beta \leq \mathbf{IN}_{j,c}$. In turn, this would imply that after the conclusion of line 13, $Clause_i = \emptyset$.

(Only If) If $Clause_i = \emptyset$ at line 14 of the procedure of figure 2.2, then for all $j \in \{1, 2, \ldots, n\}$, $\beta \leq \mathbf{IN}_{j,c}$ at line 6. Therefore, $\widehat{\mathbf{m}}_i \leq \mathbf{OUT}_{\bullet,c}$. □

To illustrate observation 2.3.2, we note that for the PN structure N_2 shown in figure 2.1(b), $\mathbf{OUT}_{\bullet,6} = (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0)^T$, which is the fourteenth minimal element of $\Delta(N_2)$ in the list shown in figure 2.1(c). If we were to run the procedure of figure 2.2 on N_2 for the transition t_6, as a consequence of observation 2.3.2, we would have $Clause_{14} = \emptyset$, which is easily verified.

If $\mathbf{OUT}_{\bullet,c} \geq \widehat{\mathbf{m}}_i$, then $\forall \mathbf{m}^1$ such that $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$, $\mathbf{m}^2 \in \Delta(N)$, this in turn implies that the condition of lemma 2.3.1 would be satisfied for $t_c \in T$. Stated in the contrapositive, if the procedure of figure 2.2 is applied only to those controllable transitions that violate the requirement of lemma 2.3.1, $Clause_i \neq \emptyset$ at line 14, $\forall i \in \{1, 2, \ldots, k\}$.

Theorem 2.3.3. *Let $N = (\Pi, T, \Phi, \Gamma)$ be a PN structure where*

$$\Delta(N) = \{\mathbf{m}^0 \in \mathcal{N}^{card(\Pi)} \mid \exists \text{ an LESP for } N(\mathbf{m}^0)\}$$

is right-closed, and $T = T_c \cup T_u$ $(T_c \cap T_u = \emptyset)$. *For each controllable transition* $t_c \in T_c$ *that does not meet the requirement of lemma 2.3.1, let* $\Theta_{t_c}(N)$ *denote the DNF expression that results when the procedure of figure 2.2 is applied to the PN structure* N *and* t_c. *Let* $\mathbf{m}^0 \in \Delta(N)$, *the supervisory policy that control-enables a* $t_c \in T_c$ *if and only if* $\Theta_{t_c}(N)$ *is true at a given marking, is equivalent to the minimally restrictive supervisory policy* $\mathcal{P}_N^*$.

Proof. As a consequence of the discussion that accompanied observation 2.3.2, the DNF-expression for $\Theta_{t_c}(N)$ does not have any empty clauses.

Suppose $\mathbf{m}^1 \in \Delta(N)$ and $\Theta_{t_c}(N)$ is true at $\mathbf{m}^1$, and $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$. It follows that $\mathbf{m}^1 \geq \mathbf{IN}_{\bullet,c}$, where $\mathbf{IN}_{\bullet,c}$ is the c-th column of the input matrix $\mathbf{IN}$.

Suppose $Clause_i$ in the DNF-expression is true at marking $\mathbf{m}^1$.

If there is a literal of the form $(\mathbf{m}(p_j) \geq \beta)$ in $Clause_i$, then $\mathbf{m}^1(p_j) \geq \max\{0, \widehat{\mathbf{m}}_i(p_j) + \mathbf{IN}_{j,c} - \mathbf{OUT}_{j,c}\} (> \mathbf{IN}_{j,c})$, and since $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$, it follows that $\mathbf{m}^2(p_j) > \widehat{\mathbf{m}}_i(p_j)$.

If there is no literal of the form $(\mathbf{m}(p_j) \geq \beta)$ in $Clause_i$, then $\beta = \max\{0, \widehat{\mathbf{m}}_i(p_j) + \mathbf{IN}_{j,c} - \mathbf{OUT}_{j,c}\} \leq \mathbf{IN}_{j,c} \Rightarrow \mathbf{OUT}_{j,c} \geq \widehat{\mathbf{m}}_i(p_j)$. Since $\mathbf{m}^2(p_j) = \mathbf{m}^1(p_j) - \mathbf{IN}_{j,c} + \mathbf{OUT}_{j,c}$, and $\mathbf{m}^1(p_j) \geq \mathbf{IN}_{j,c}$, it follows that $\mathbf{m}^2(p_j) \geq \mathbf{OUT}_{j,c} \geq \widehat{\mathbf{m}}_i(p_j)$.

Therefore, $\mathbf{m}^2 \geq \widehat{\mathbf{m}}_i \Rightarrow \mathbf{m}^2 \in \Delta(N)$. Consequently, $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$ under the supervision of $\mathcal{P}_N^*$ too.

If $\mathbf{m}^1 \xrightarrow{t_c} \mathbf{m}^2$ under the supervision of $\mathcal{P}_N^*$, then $\exists \widehat{\mathbf{m}}_i$ such that $\mathbf{m}^2 \geq \widehat{\mathbf{m}}_i$. Since, $\mathbf{m}^2 = \mathbf{m}^1 - \mathbf{IN}_{\bullet,c} + \mathbf{OUT}_{\bullet,c} \geq \widehat{\mathbf{m}}_i \Rightarrow \mathbf{m}^1 \geq \widehat{\mathbf{m}}_i + \mathbf{IN}_{\bullet,c} - \mathbf{OUT}_{\bullet,c}$, where $\mathbf{OUT}_{\bullet,c}$ is the c-th column of the output matrix $\mathbf{OUT}$. Consequently, if there is a literal of the form $(\mathbf{m}(p_j) \geq \beta)$ in $Clause_i$, it will be satisfied by $\mathbf{m}^1$. Therefore, $\Theta_{t_c}(N)$ will be satisfied by $\mathbf{m}^1$. □

If any marking that satisfies $Clause_j$ also satisfies $Clause_i$, then $Clause_j$ can be eliminated from the DNF-expression for $\Theta_{t_c}(N)$. This would be the case if $Clause_i$ is a sub-clause of $Clause_j$ in the DNF-expression for $\Theta_{t_c}(N)$; or, if $Clause_j$ and $Clause_i$ are atomic clauses of the form "$\mathbf{m}(p) \geq \beta_1$" and "$\mathbf{m}(p) \geq \beta_2$" respectively, where $\beta_2 < \beta_1$.

CHAPTER 3

EXAMPLES

For a given PN structure N, there are two supervisory policies discussed throughout this chapter. The first supervisory policy $\mathcal{P}_1$ makes use of the DNF expression that will fire the controllable transition $t_c \in T_c$ at a marking $\mathbf{m}$ if and only if the DNF expression evaluates to be "True" at $\mathbf{m}$. The second supervisory policy $\mathcal{P}_2$ (which is the *minimally restrictive policy*) permits the firing of the controllable transition $t_c \in T_c$ at a marking $\mathbf{m}$ if and only if the marking $\widehat{\mathbf{m}}$ that results from the firing of t_c is in the right closed set $\Delta(N)$. It will be shown that for each example, the two policies $\mathcal{P}_1$ and $\mathcal{P}_2$ work identically. Since the two policies are identical, $\mathcal{P}_1$ is also found to be minimally restrictive.

3.1 Example 1

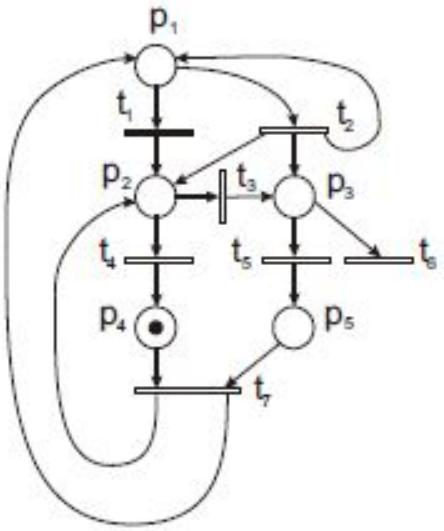


Figure 3.1: Petri net N_1.

```
(Final) Minimal Elements of the control-invariant set
-----------------------------------------------------

 1: ( 1 0 0 0 0 )
 2: ( 0 0 0 1 1 )
```

Figure 3.2: Minimal element set $min(\Delta(N_1))$.

```
{m(p(1))>=2} ||{m(p(4))>=1 && m(p(5))>=1}
```

Figure 3.3: The DNF expression $\Theta_{t_1}(N_1)$.

Transition t_1 is the only controllable transition in the PN N_1 shown in figure 3.1. The minimal elements of $\Delta(N_1)$ are $\widehat{\mathbf{m}}_1 = (1\ 0\ 0\ 0\ 0)^T$ and $\widehat{\mathbf{m}}_2 = (0\ 0\ 0\ 1\ 1)^T$, as shown in figure 3.2. The DNF expression for the Petri net, $\Theta_{t_1}(N_1)$ is determined by applying the algorithm in the previous chapter. The final DNF expression is shown in figure 3.3. The first clause $\big(\mathbf{m}(p_1) \geq 2\big)$ indicates that it is sufficient if a 2-threshold sensor is at place p_1. Similarly, the second clause $\big((\mathbf{m}(p_4) \geq 1) \wedge (\mathbf{m}(p_5) \geq 1)\big)$ indicates that it is sufficient if 1-threshold sensors are at places p_4 and p_5. We will show that the supervisory policies $\mathcal{P}_1$ and $\mathcal{P}_2$, which are defined at introduction to this chapter, are identical.

If the DNF expression $\Theta_{t_1}(N_1)$ is "False" for some marking $\mathbf{m}$, then the policy $\mathcal{P}_1$ will not permit the firing of t_1 at marking $\mathbf{m}$. Since $\theta_{t_1}(N_1)$ is "False," it follows that $\big(\mathbf{m}(p_1) \leq 1\big) \wedge \big((\mathbf{m}(p_4) = 0) \vee (\mathbf{m}(p_5) = 0)\big)$. If t_1 is to fire at marking $\mathbf{m}$, it must be that $\mathbf{m}(p_1) = 1$. The resultant marking after the firing of t_1 will be $(0\ (1+\mathbf{m}(p_2))\ \mathbf{m}(p_3)\ \mathbf{m}(p_4)\ \mathbf{m}(p_5))^T$. This marking is not in the right closed set, $\Delta(N_1)$, as $\big((\mathbf{m}(p_4) = 0) \vee (\mathbf{m}(p_5) = 0)\big)$. Consequently, the policy $\mathcal{P}_2$ will not permit the firing of t_1 at $\mathbf{m}$ either.

If the clauses $\big(\mathbf{m}(p_1) \geq 2\big)$ or $\big((\mathbf{m}(p_4) \geq 1) \wedge (\mathbf{m}(p_5) \geq 1)\big)$ are "True" at a marking $\mathbf{m}$, the policy $\mathcal{P}_1$ will permit its firing at $\mathbf{m}$. If t_1 can be fired at $\mathbf{m}$, the new marking that would result will be greater than $\widehat{\mathbf{m}}_1$ or $\widehat{\mathbf{m}}_2$. The policy $\mathcal{P}_2$ will also permit the firing of t_1 at marking $\mathbf{m}$.

Therefore, the policy obtained by DNF expression $\mathcal{P}_1$ and the minimally restrictive policy $\mathcal{P}_2$ are one and the same.

3.2 Example 2

Figure 3.4 shows that transition t_1 is the only controllable transition of the PN N_2. The minimal elements of $\Delta(N_2)$ are $\widehat{\mathbf{m}}_1 = (1\ 0\ 0\ 0\ 0)^T$ and $\widehat{\mathbf{m}}_2 = (0\ 0\ 0\ 1\ 1)^T$, as shown in figure 3.5. After applying the algorithm in the previous section, the DNF expression, $\Theta_{t_1}(N_2)$ is determined. The final DNF expression is shown in figure 3.6. The first clause $\big(\mathbf{m}(p_1) \geq 3\big)$ indicates that it is sufficient if a 3-threshold sensor is at place p_1. Similarly, the second clause $\big((\mathbf{m}(p_4) \geq 1) \wedge (\mathbf{m}(p_5) \geq 1)\big)$ indicates that it is sufficient if 1-threshold sensors are at places p_4 and p_5.

If the DNF expression $\Theta_{t_1}(N_2)$ was "False" at a marking $\mathbf{m}$, then $\mathcal{P}_1$ will not permit its firing at $\mathbf{m}$, and $\big(\mathbf{m}(p_1) \leq 2\big) \wedge \big((\mathbf{m}(p_4) = 0) \vee (\mathbf{m}(p_5) = 0)\big)$. If t_1 can fire at $\mathbf{m}$, it must be that $\mathbf{m}(p_1) = 2$. The resultant marking after the firing of t_1 at $\mathbf{m}$ will be $(0\ (1 + \mathbf{m}(p_2))\ \mathbf{m}(p_3)\ \mathbf{m}(p_4)\ \mathbf{m}(p_5))^T$. This marking is not in the right closed set, $\Delta(N_2)$, as $\big((\mathbf{m}(p_4) = 0) \vee (\mathbf{m}(p_5) = 0)\big)$. So, the policy $\mathcal{P}_2$ will not permit the firing of t_1 at $\mathbf{m}$ either.

If the clauses $\big(\mathbf{m}(p_1) \geq 3\big)$ or $\big((\mathbf{m}(p_4) \geq 1) \wedge (\mathbf{m}(p_5) \geq 1)\big)$ are true and p_1 has enough tokens to fire t_1, the new marking that would result from the firing of t_1 at $\mathbf{m}$ will be greater than $\widehat{\mathbf{m}}_1$ or $\widehat{\mathbf{m}}_2$. The policy $\mathcal{P}_2$ will also permit the firing of t_1 at marking $\mathbf{m}$.

Therefore, the minimally restrictive policy $\mathcal{P}_2$ and the policy obtained by DNF expression $\mathcal{P}_1$ are one and the same.

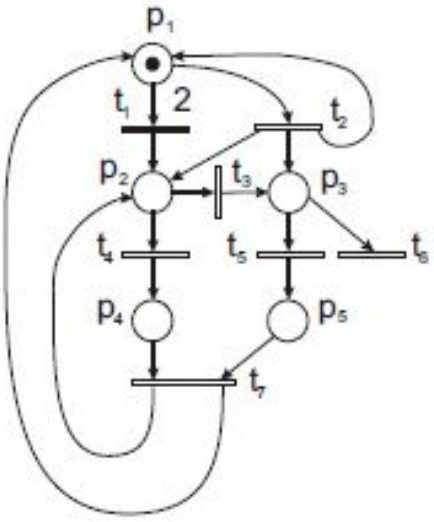


Figure 3.4: Petri net N_2.

```
(Final) Minimal Elements of the control-invariant set
-----------------------------------------------------

 1: ( 1 0 0 0 0 )
 2: ( 0 0 0 1 1 )
```

Figure 3.5: Minimal element set $min(\Delta(N_2))$.

```
{m(p(1))>=3} ||{m(p(4))>=1 && m(p(5))>=1}
```

Figure 3.6: DNF expression $\Theta_{t_1}(N_2)$.

3.3 Example 3

Figure 3.7 shows that transition t_1 is the only controllable transition of the PN N_3. The minimal elements of $\Delta(N_3)$ are as shown in figure 3.8. After applying the algorithm in the previous section, the DNF expression, $\Theta_{t_1}(N_3)$ is determined. The final DNF expression is shown in figure 3.9. The first clause $\big(\mathbf{m}(p_1) \geq 2\big)$ indicates that it is sufficient if there is a 2-threshold sensor at place p_1. The second clause $\big(\mathbf{m}(p_4) \geq 1\big)$, third clause $\big(\mathbf{m}(p_6) \geq 1\big)$, fourth clause $\big(\mathbf{m}(p_7) \geq 1\big)$ and final clause $\big(\mathbf{m}(p_8) \geq 1\big)$ indicate that it is sufficient if 1-threshold sensors are at places p_4, p_6, p_7 and p_8.

If the DNF expression $\Theta_{t_1}(N_3)$ is "False" at a marking $\mathbf{m}$, then $\mathcal{P}_1$ will not permit its firing at $\mathbf{m}$, and $\big(\mathbf{m}(p_1) \leq 1\big) \wedge \big(\mathbf{m}(p_4) = 0\big) \wedge \big(\mathbf{m}(p_6) = 0\big) \wedge \big(\mathbf{m}(p_7) = 0\big) \wedge \big(\mathbf{m}(p_8) = 0\big)$. If t_1 can fire at $\mathbf{m}$, it must be that $\mathbf{m}(p_1) = 1$. The resultant marking after the firing of t_1 at $\mathbf{m}$ will be $(0\ (1 + \mathbf{m}(p_2))\ \mathbf{m}(p_3)\ 0\ \mathbf{m}(p_5)\ 0\ 0\ 0)^T$. This marking is not in the right closed set, $\Delta(N_3)$, as the resultant marking does not have any token load in places p_1, p_4, p_6, p_7 and p_8. Therefore the resultant marking is not greater than or equal to any of the members of $\Delta(N_3)$. So, the policy $\mathcal{P}_2$ will not permit the firing of t_1 at $\mathbf{m}$ either.

If the clauses $\big(\mathbf{m}(p_1) \geq 2\big)$ or $\big(\mathbf{m}(p_4) \geq 1\big)$ or $\big(\mathbf{m}(p_6) \geq 1\big)$ or $\big(\mathbf{m}(p_7) \geq 1\big)$ or $\big(\mathbf{m}(p_8) \geq 1\big)$ are true and p_1 has enough tokens to fire t_1, the new marking that would result from the firing of t_1 at $\mathbf{m}$ will be greater than one of the minimal members of $\Delta(N_3)$. The policy $\mathcal{P}_2$ will also permit the firing of t_1 at marking $\mathbf{m}$.

Therefore, the minimally restrictive policy $\mathcal{P}_2$ and the policy obtained by DNF expression $\mathcal{P}_1$ are one and the same.

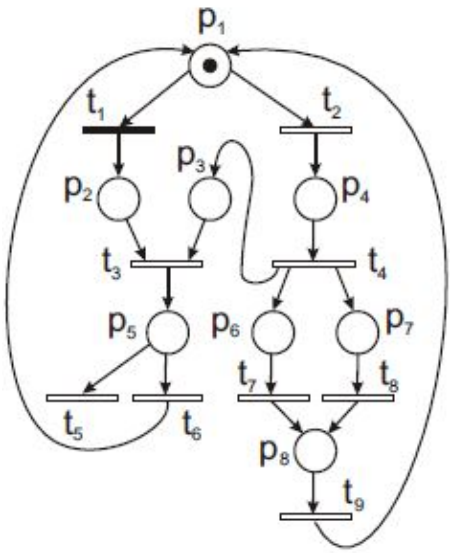


Figure 3.7: Petri net N_3.

```
(Final) Minimal Elements of the control-invariant set
------------------------------------------------------

 1: ( 1 0 0 0 0 0 0 0 0 )
 2: ( 0 0 0 1 0 0 0 0 0 )
 3: ( 0 0 0 0 0 0 1 0 0 )
 4: ( 0 0 0 0 0 0 0 1 0 )
 5: ( 0 0 0 0 0 0 0 0 1 )
```

Figure 3.8: Minimal element set $min(\Delta(N_3))$.

```
{m(p(1))>=2}
||
{m(p(4))>=1}
||
{m(p(6))>=1}
||
{m(p(7))>=1}
||
{m(p(8))>=1}
```

Figure 3.9: The DNF expression $\Theta_{t_1}(N_3)$.

3.4 Example 4

In Figure 3.10, it is seen that t_4 is the only controllable transition of the PN N_4. The minimal elements of $\Delta(N_4)$ are as shown in figure 3.11. The DNF expression $\Theta_{t_4}(N_4)$ is shown in figure 3.12. As discussed in the previous chapter, if two clauses $clause_i$ and $clause_j$ exist such that $clause_i$ is a sub-clause of $clause_j$ or $clause_j$ and $clause_i$ are of the form "$\mathbf{m}(p) \geq \beta_1$" and "$\mathbf{m}(p) \geq \beta_2$" where $\beta_2 < \beta_1$, provided the marking that satisfies both clauses exists, then $clause_j$ can be eliminated from the DNF expression. Based on this observation, the clauses of the DNF expression reduces to $\big(\mathbf{m}(p_1) \geq 1\big) \vee \big((\mathbf{m}(p_2) \geq 1) \wedge (\mathbf{m}(p_4) \geq 2)\big) \vee \big((\mathbf{m}(p_2) \geq 1) \wedge (\mathbf{m}(p_3) \geq 1)\big)$. The first clause $\big(\mathbf{m}(p_1) \geq 1\big)$ indicates that it is sufficient if a 1-threshold sensor is at place p_1. The second clause $\big(\mathbf{m}(p_2) \geq 1\big) \wedge \big(\mathbf{m}(p_4) \geq 2\big)$ and third clause $\big((\mathbf{m}(p_2) \geq 1) \wedge (\mathbf{m}(p_3) \geq 1)\big)$ indicate that it is sufficient if there are 1-threshold sensors at p_1, p_2 and p_3 and a 2-threshold sensor at p_4.

If the DNF expression $\Theta_{t_4}(N_4)$ is "False" at a marking $\mathbf{m}$, then $\mathcal{P}_1$ will not permit its firing at $\mathbf{m}$, and $\big(\mathbf{m}(p_1) = 0\big) \wedge \big((\mathbf{m}(p_2) = 0) \vee (\mathbf{m}(p_4) \leq 1)\big) \wedge \big((\mathbf{m}(p_2) = 0) \vee (\mathbf{m}(p_3) = 0)\big)$. If t_4 can fire at $\mathbf{m}$, it must be that $\mathbf{m}(p_4) = 1$. The resultant marking after the firing of t_4 at $\mathbf{m}$ will be $(0\ \mathbf{m}(p_2)\ \mathbf{m}(p_3)\ 0\ (1 + \mathbf{m}(p_5)))^T$. The only minimal member of $\Delta(N_4)$ where $\mathbf{m}(p_5) = 1$ also has the condition that p_1 has unity token load. Therefore, the resultant marking is not in right closed set, $\Delta(N_4)$. So, the policy $\mathcal{P}_2$ will not permit the firing of t_4 at $\mathbf{m}$ either.

If the clauses $\big(\mathbf{m}(p_1) \geq 1\big)$ or $\big((\mathbf{m}(p_2) \geq 1) \wedge (\mathbf{m}(p_4) \geq 2)\big)$ or $\big((\mathbf{m}(p_2) \geq 1) \wedge (\mathbf{m}(p_3) \geq 1)\big)$ are true and p_4 has enough tokens to fire t_4, the new marking that would result from the firing of t_4 at $\mathbf{m}$ will be greater than one of the minimal members of $\Delta(N_4)$. The policy $\mathcal{P}_2$ will also permit the firing of t_4 at marking $\mathbf{m}$.

Therefore, the minimally restrictive policy $\mathcal{P}_2$ and the policy obtained by DNF expression $\mathcal{P}_1$ are one and the same.

3.5 Example 5

From Figure 3.13, we can infer that t_3 is the only controllable transition of the PN N_5. The minimal elements of $\Delta(N_5)$ are as shown in figure 3.14. The DNF expression obtained from the algorithm, $\Theta_{t_3}(N_5)$ is determined and shown in figure 3.15. The first clause $\big(\mathbf{m}(p_2) \geq 4\big)$ and and second clause $\big(\mathbf{m}(p_1) \geq 1\big)$ indicate that it is sufficient if a 4-threshold sensor is at place p_2 and a 1-threshold sensor is at place p_1.

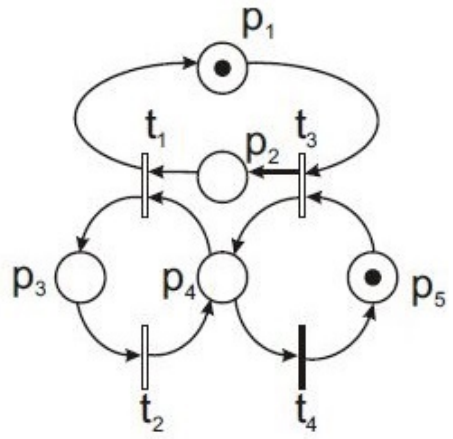


Figure 3.10: Petri net N_4.

```
(Final) Minimal Elements of the control-invariant set
------------------------------------------------------

 1: ( 1 0 0 0 1 )
 2: ( 1 0 0 1 0 )
 3: ( 0 1 0 1 0 )
 4: ( 1 0 1 0 0 )
 5: ( 0 1 1 0 0 )
```

Figure 3.11: Minimal element set $min(\Delta(N_4))$.

```
{m(p(1))>=1}
||
{m(p(1))>=1 && m(p(4))>=2}
||
{m(p(2))>=1 && m(p(4))>=2}
||
{m(p(1))>=1 && m(p(3))>=1}
||
{m(p(2))>=1 && m(p(3))>=1}
```

Figure 3.12: The DNF expression $\Theta_{t_4}(N_4)$.

If the DNF expression $\Theta_{t_3}(N_5)$ is "False" at a marking $\mathbf{m}$, then $\mathcal{P}_1$ will not permit its firing at $\mathbf{m}$, and $\big(\mathbf{m}(p_2) \leq 3\big) \wedge \big(\mathbf{m}(p_1) = 0\big)$. If t_3 can fire at $\mathbf{m}$, it must be that $\mathbf{m}(p_2)$ must be either 2 or 3. The resultant marking after the firing of t_3 at $\mathbf{m}$ when $\mathbf{m}(p_2) = 2$ will be $(1\ 0)^T$ and when $\mathbf{m}(p_2) = 3$ is $(1\ 1)^T$. These markings are not in the right closed set, $\Delta(N_5)$. So, the policy $\mathcal{P}_2$ will not permit the firing of t_3 at $\mathbf{m}$ either.

If the clauses $\big(\mathbf{m}(p_2) \geq 4\big)$ or $\big(\mathbf{m}(p_1) \geq 1\big)$ are true and p_2 has enough tokens to fire t_3, the new marking that would result from the firing of t_3 at $\mathbf{m}$ will be greater than some minimal member of $\Delta(N_5)$. The policy $\mathcal{P}_2$ will also permit the firing of t_3 at marking $\mathbf{m}$.

Therefore, the minimally restrictive policy $\mathcal{P}_2$ and the policy obtained by DNF expression $\mathcal{P}_1$ are the same.

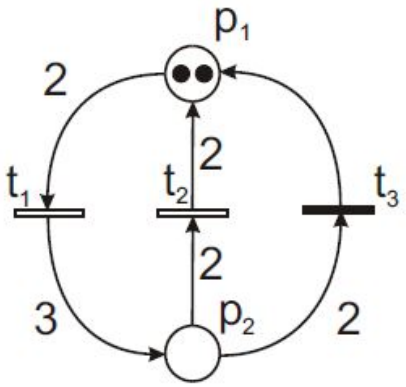


Figure 3.13: Petri net N_5.

```
(Final) Minimal Elements of the control-invariant set
------------------------------------------------------

 1: ( 0 2 )
 2: ( 2 0 )
```

Figure 3.14: Minimal element set $min(\Delta(N_5))$.

```
{m(p(2))>=4}
||
{m(p(1))>=1}
```

Figure 3.15: The DNF expression $\Theta_{t_3}(N_5)$.

3.6 Example 6

Figure 3.16 shows that transition t_5 is the only controllable transition of the PN N_6. The minimal elements of $\Delta(N_6)$ are as shown in figure 3.17. After applying the algorithm in the previous section, the DNF expression, $\Theta_{t_5}(N_6)$ is determined. The final DNF expression is in the form shown in figure 3.18. Since $\big(\mathbf{m}(p_1) \geq 1\big)$ and $\big(\mathbf{m}(p_1) \geq 2\big)$ are atomic clauses of the form "$\mathbf{m}(p) \geq \beta_1$" and "$\mathbf{m}(p) \geq \beta_2$" where $\beta_2 > \beta_1$, $\big(\mathbf{m}(p_1) \geq 2\big)$ is redundant clause due its higher threshold and therefore eliminated. The first clause $\big(\mathbf{m}(p_3) \geq 1\big)$, second clause $\big(\mathbf{m}(p_1) \geq 1\big)$, and third clause $\big(\mathbf{m}(p_4) \geq 1\big)$ indicate that it is sufficient if a 1-threshold sensor is at places p_3, p_1 and p_4 respectively. The final clause $\big(\mathbf{m}(p_2) \geq 4\big)$ indicates that it is sufficient if a 4-threshold sensor is at place p_2.

If the DNF expression $\Theta_{t_5}(N_6)$ is "False" at a marking $\mathbf{m}$, then $\mathcal{P}_1$ will not permit its firing at $\mathbf{m}$, and $\big(\mathbf{m}(p_3) = 0\big) \wedge \big(\mathbf{m}(p_1) = 0\big) \wedge \big(\mathbf{m}(p_4) = 0\big) \wedge \big(\mathbf{m}(p_2) \leq 3\big)$. If t_5 can fire at $\mathbf{m}$, it must be that $\mathbf{m}(p_2)$ is either 2 or 3. The resultant marking after the firing of t_5 at $\mathbf{m}$ when $\mathbf{m}(p_2) = 2$ will be $(0\ 0\ 0\ 1\ )^T$. When $\mathbf{m}(p_2) = 3$, the resultant marking after the firing of t_5 at $\mathbf{m}$ will be $(0\ 1\ 0\ 1\ )^T$. These markings are not in the right closed set, $\Delta(N_6)$. So, the policy $\mathcal{P}_2$ will not permit the firing of t_5 at $\mathbf{m}$ either.

If the clauses $\big(\mathbf{m}(p_3) \geq 1\big)$ or $\big(\mathbf{m}(p_1) \geq 1\big)$ or $\big(\mathbf{m}(p_4) \geq 1\big)$ or $\big(\mathbf{m}(p_2) \geq 4\big)$ are "True" and p_2 has enough tokens to fire t_5, the new marking that would result from the firing of t_5 at $\mathbf{m}$ will be greater than some minimal member of $\Delta(N_6)$. The policy $\mathcal{P}_2$ will also permit the firing of t_5 at marking $\mathbf{m}$. Therefore, the minimally restrictive policy $\mathcal{P}_2$ and the policy obtained by DNF expression $\mathcal{P}_1$ are the same.

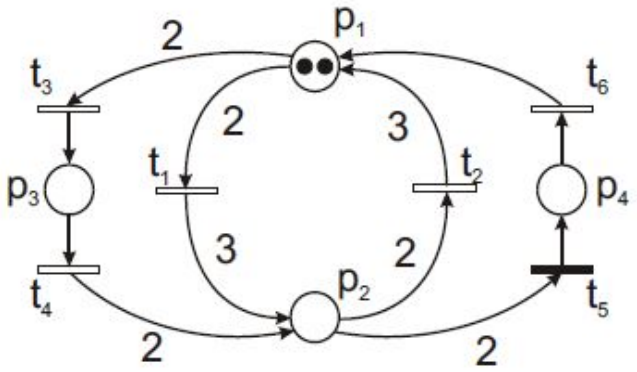


Figure 3.16: Petri net N_6.

```
(Final) Minimal Elements of the control-invariant set
-----------------------------------------------------

 1: ( 0 0 1 0 )
 2: ( 1 0 0 1 )
 3: ( 0 0 0 2 )
 4: ( 0 2 0 0 )
 5: ( 2 0 0 0 )
```

Figure 3.17: Minimal element set $min(\Delta(N_6))$.

```
{m(p(3))>=1}
||
{m(p(1))>=1}
||
{m(p(4))>=1}
||
{m(p(2))>=4}
||
{m(p(1))>=2}
```

Figure 3.18: The DNF expression $\Theta_{t_5}(N_6)$.

3.7 Example 7

Transitions t_1, t_2, t_3 and t_4 are the controllable transitions that of the PN N_7 from Figure 3.19 being considered. The minimal elements of $\Delta(N_7)$ are as shown in figure 3.20. The DNF expressions for transitions t_1, t_2, t_3 and t_4, obtained from the algorithm in the previous section, $\Theta_{t_1}(N_7)$, $\Theta_{t_2}(N_7)$, $\Theta_{t_3}(N_7)$ and $\Theta_{t_4}(N_7)$ are shown in figures 3.21, 3.22, 3.23 and 3.24. Transition t_4, like transitions t_5, t_6, t_8, t_9, t_{10} and t_{11}, satisfies lemma, and will never be disabled by $\mathcal{P}_2$. Later, we will show that DNF formula $\Theta_{t_4}(N_7)$ will always be true and t_4 will always be control enabled by policy $\mathcal{P}_1$.

The final DNF expression of $\Theta_{t_1}(N_7)$ is shown in figure 3.21. After eliminating the redundant clauses from the DNF expressions using the observations made in the previous section, $\Theta_{t_1}(N_7)$ reduces to $\big(\mathbf{m}(p_4) \geq 1\big) \vee \big(\mathbf{m}(p_1) \geq 2\big) \vee \big(\mathbf{m}(p_7 \geq 1)\big) \vee \big(\mathbf{m}(p_8) \geq 1\big) \vee \big(\mathbf{m}(p_9) \geq 1\big)$. Similarly $\Theta_{t_3}(N_7)$ and $\Theta_{t_4}(N_7)$ reduce to $\big(\mathbf{m}(p_5) \geq 1\big) \vee \big(\mathbf{m}(p_2) \geq 1\big) \vee \big(\mathbf{m}(p_1) \geq 2\big) \vee \big(\mathbf{m}(p_7) \geq 1\big) \vee \big(\mathbf{m}(p_8) \geq 1\big) \vee \big(\mathbf{m}(p_9) \geq 1\big)$ and $\big(\mathbf{m}(p_1) \geq 1\big) \vee \big(\mathbf{m}(p_4) \geq 1\big) \vee \big(\mathbf{m}(p_7) \geq 1\big) \vee \big(\mathbf{m}(p_8) \geq 1\big) \vee \big(\mathbf{m}(p_9) \geq 1\big)$ respectively. $\Theta_{t_2}(N_7)$ does not have any redundant clauses and therefore the DNF expression remains the same.

According to $\Theta_{t_1}(N_7)$, the clauses indicate that it is sufficient if 1-threshold sensors are placed at p_4, p_7, p_8 and p_9 and a 2-threshold sensor at p_1.

If the DNF expression $\Theta_{t_1}(N_7)$ is "False" at a marking $\mathbf{m}$, then $\big(\mathbf{m}(p_4) = 0\big) \wedge \big(\mathbf{m}(p_1) \leq$

$1\big) \wedge \big(\mathbf{m}(p_7) = 0\big) \wedge \big(\mathbf{m}(p_8) = 0\big) \wedge \big(\mathbf{m}(p_9) = 0\big)$ and $\mathcal{P}_1$ will not permit its firing at $\mathbf{m}$. In order for t_1 to fire, p_1 should have a token load of one. The resultant marking after the firing of t_1 at $\mathbf{m}$ will be $(0\ (1+\mathbf{m}(p_2))\ \mathbf{m}(p_3)\ 0\ \mathbf{m}(p_5)\ \mathbf{m}(p_6)\ 0\ 0\ 0)^T$. The only two minimal elements where p_1, p_7, p_8 and p_9 have zero token load is $\widehat{\mathbf{m}}_1$ and $\widehat{\mathbf{m}}_2$ (cf. Figure 3.20). Since $\mathbf{m}(p_4) = 0$, these minimal elements are also eliminated. Hence, the resultant marking is not in $\Delta(N_7)$ and the policy $\mathcal{P}_2$ will not permit the firing of t_1 at $\mathbf{m}$.

If the DNF expression is "True," and the number of tokens in p_1 are sufficient to fire t_1, then the resultant marking is in the right closed set, $\Delta(N_7)$. The policy $\mathcal{P}_2$ will also permit the firing of t_1 at marking $\mathbf{m}$.

In a similar manner to $\Theta_{t_1}(N_7)$, the clauses of $\Theta_{t_2}(N_7)$ can be used to determine the threshold sensors needed for the places of PN N_7. It can be shown that the resultant marking, from the firing of t_2 at any marking that makes all clauses of the DNF "False," for $\Theta_{t_2}(N_7)$, is not in $\Delta(N_7)$. Therefore, the policy $\mathcal{P}_2$ will not permit the firing of t_2 at marking $\mathbf{m}$. When the DNF expression is found to be "True," the resultant marking is in $\Delta(N_7)$, and policy $\mathcal{P}_2$ would permit its firing. This is not elaborated in interest of space.

In the case of $\Theta_{t_3}(N_7)$, the reduced DNF expression suggests that it is sufficient if 1-threshold sensors are placed at p_5, p_2, p_1, p_7, p_8 and p_9. If the DNF expression is "False," the resultant marking obtained is $(0\ 0\ (1+\mathbf{m}(p_3))\ (1+\mathbf{m}(p_4))\ 0\ \mathbf{m}(p_6)\ 0\ 0\ 0)^T$ because token load at p_1 has to be 1 to fire t_3. Using a similar argument for the resultant marking from the firing of t_1 described in the previous case, we can show that the resultant marking in this case is not in $\Delta(N_7)$. Hence, policy $\mathcal{P}_2$ will not allow the firing of t_3 at $\mathbf{m}$. On the other hand, if the reduced DNF expression is "True" (i.e.) $\big(\mathbf{m}(p_4) \geq 1\big) \vee \big(\mathbf{m}(p_1) \geq 2\big) \vee \big(\mathbf{m}(p_7) \geq 1\big) \vee \big(\mathbf{m}(p_8) \geq 1\big) \vee \big(\mathbf{m}(p_9) \geq 1\big)$, then the resultant marking is found to be greater than one of the minimal members of $\Delta(N_7)$, provided there are sufficient number of tokens in p_1.

The reduced DNF expression $\Theta_{t_4}(N_7)$ is $\big(\mathbf{m}(p_1) \geq 1\big) \vee \big(\mathbf{m}(p_4) \geq 1\big) \vee \big(\mathbf{m}(p_7) \geq 1\big) \vee \big(\mathbf{m}(p_8) \geq 1\big) \vee \big(\mathbf{m}(p_9) \geq 1\big)$. It is seen that the reduced DNF formula expressed above is "True" for each of the twenty-four minimal elements of $\Delta(N_7)$ (shown in Figure 3.20). Since any member of $\Delta(N_7)$ is greater than or equal to at least one of these twenty-four elements, the reduced DNF expression $\Theta_{t_4}(N_7)$ is "True" for any member of $\Delta(N_7)$. Thus, the controllable transition t_4 is always found to be control enabled by the policy $\mathcal{P}_1$.

Since the "False" DNF expressions correspond to markings which are not in the right closed set and "True" DNF expressions correspond to markings which are in the right closed set, the minimally restrictive policy $\mathcal{P}_2$ and the policy obtained by DNF expression $\mathcal{P}_1$ are

the same.

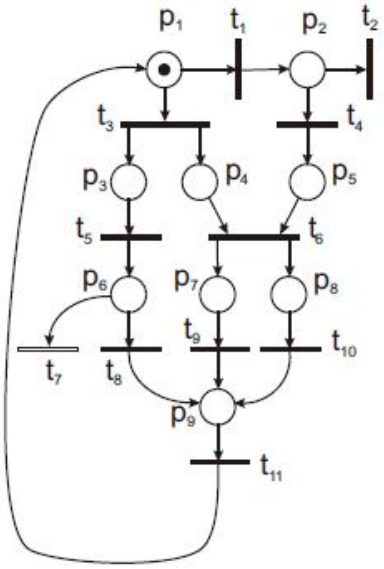


Figure 3.19: Petri net N_7.

```
(Final) Minimal Elements of the control-invariant set
------------------------------------------------------

 1: ( 0 0 0 1 1 0 0 0 0 )
 2: ( 0 1 0 1 0 0 0 0 0 )
 3: ( 2 0 0 0 0 0 0 0 0 )
 4: ( 1 1 0 0 0 0 0 0 0 )
 5: ( 1 0 0 1 0 0 0 0 0 )
 6: ( 1 0 0 0 1 0 0 0 0 )
 7: ( 1 0 0 0 0 0 1 0 0 )
 8: ( 1 0 0 0 0 0 0 1 0 )
 9: ( 1 0 0 0 0 0 0 0 1 )
10: ( 0 1 0 0 0 0 1 0 0 )
11: ( 0 0 0 1 0 0 1 0 0 )
12: ( 0 0 0 0 1 0 1 0 0 )
13: ( 0 0 0 0 0 0 2 0 0 )
14: ( 0 0 0 0 0 0 1 1 0 )
15: ( 0 0 0 0 0 0 1 0 1 )
16: ( 0 1 0 0 0 0 0 1 0 )
17: ( 0 0 0 1 0 0 0 1 0 )
18: ( 0 0 0 0 1 0 0 1 0 )
19: ( 0 0 0 0 0 0 0 2 0 )
20: ( 0 0 0 0 0 0 0 1 1 )
21: ( 0 1 0 0 0 0 0 0 1 )
22: ( 0 0 0 1 0 0 0 0 1 )
23: ( 0 0 0 0 1 0 0 0 1 )
24: ( 0 0 0 0 0 0 0 0 2 )
```

Figure 3.20: Minimal element set $min(\Delta(N_7))$.

```
(m(p(4))>=1 && m(p(5))>=1)
||
(m(p(4))>=1)
||
(m(p(1))>=3)
||
(m(p(1))>=2)
||
(m(p(1))>=2 && m(p(4))>=1)
||
(m(p(1))>=2 && m(p(5))>=1)
||
(m(p(1))>=2 && m(p(7))>=1)
||
(m(p(1))>=2 && m(p(8))>=1)
||
(m(p(1))>=2 && m(p(9))>=1)
||
(m(p(7))>=1)
||
(m(p(4))>=1 && m(p(7))>=1)
||
(m(p(5))>=1 && m(p(7))>=1)
||
(m(p(7))>=2)
||
(m(p(7))>=1 && m(p(8))>=1)
||
(m(p(7))>=1 && m(p(9))>=1)
||
(m(p(8))>=1)
||
(m(p(4))>=1 && m(p(8))>=1)
||
(m(p(5))>=1 && m(p(8))>=1)
||
(m(p(8))>=2)
||
(m(p(8))>=1 && m(p(9))>=1)
||
(m(p(9))>=1)
||
(m(p(4))>=1 && m(p(9))>=1)
||
(m(p(5))>=1 && m(p(9))>=1)
||
(m(p(9))>=2)
```

Figure 3.21: The DNF expression $\Theta_{t_1}(N_7)$.

```
(m(p(4))>=1 && m(p(5))>=1)
||
(m(p(2))>=2 && m(p(4))>=1)
||
(m(p(1))>=2)
||
(m(p(1))>=1 && m(p(2))>=2)
||
(m(p(1))>=1 && m(p(4))>=1)
||
(m(p(1))>=1 && m(p(5))>=1)
||
(m(p(1))>=1 && m(p(7))>=1)
||
(m(p(1))>=1 && m(p(8))>=1)
||
(m(p(1))>=1 && m(p(9))>=1)
||
(m(p(2))>=2 && m(p(7))>=1)
||
(m(p(4))>=1 && m(p(7))>=1)
||
(m(p(5))>=1 && m(p(7))>=1)
||
(m(p(7))>=2)
||
(m(p(7))>=1 && m(p(8))>=1)
||
(m(p(7))>=1 && m(p(9))>=1)
||
(m(p(2))>=2 && m(p(8))>=1)
||
(m(p(4))>=1 && m(p(8))>=1)
||
(m(p(5))>=1 && m(p(8))>=1)
||
(m(p(8))>=2)
||
(m(p(8))>=1 && m(p(9))>=1)
||
(m(p(2))>=2 && m(p(9))>=1)
||
(m(p(4))>=1 && m(p(9))>=1)
||
(m(p(5))>=1 && m(p(9))>=1)
||
(m(p(9))>=2)
```

Figure 3.22: The DNF expression $\Theta_{t_2}(N_7)$.

```
(m(p(5))>=1)
||
(m(p(2))>=1)
||
(m(p(1))>=3)
||
(m(p(1))>=2 && m(p(2))>=1)
||
(m(p(1))>=2)
||
(m(p(1))>=2 && m(p(5))>=1)
||
(m(p(1))>=2 && m(p(7))>=1)
||
(m(p(1))>=2 && m(p(8))>=1)
||
(m(p(1))>=2 && m(p(9))>=1)
||
(m(p(2))>=1 && m(p(7))>=1)
||
(m(p(7))>=1)
||
(m(p(5))>=1 && m(p(7))>=1)
||
(m(p(7))>=2)
||
(m(p(7))>=1 && m(p(8))>=1)
||
(m(p(7))>=1 && m(p(9))>=1)
||
(m(p(2))>=1 && m(p(8))>=1)
||
(m(p(8))>=1)
||
(m(p(5))>=1 && m(p(8))>=1)
||
(m(p(8))>=2)
||
(m(p(8))>=1 && m(p(9))>=1)
||
(m(p(2))>=1 && m(p(9))>=1)
||
(m(p(9))>=1)
||
(m(p(5))>=1 && m(p(9))>=1)
||
(m(p(9))>=2)
```

Figure 3.23: The DNF expression $\Theta_{t_3}(N_7)$.

```
(m(p(4))>=1)
||
(m(p(2))>=2 && m(p(4))>=1)
||
(m(p(1))>=2)
||
(m(p(1))>=1 && m(p(2))>=2)
||
(m(p(1))>=1 && m(p(4))>=1)
||
(m(p(1))>=1)
||
(m(p(1))>=1 && m(p(7))>=1)
||
(m(p(1))>=1 && m(p(8))>=1)
||
(m(p(1))>=1 && m(p(9))>=1)
||
(m(p(2))>=2 && m(p(7))>=1)
||
(m(p(4))>=1 && m(p(7))>=1)
||
(m(p(7))>=1)
||
(m(p(7))>=2)
||
(m(p(7))>=1 && m(p(8))>=1)
||
(m(p(7))>=1 && m(p(9))>=1)
||
(m(p(2))>=2 && m(p(8))>=1)
||
(m(p(4))>=1 && m(p(8))>=1)
||
(m(p(8))>=1)
||
(m(p(8))>=2)
||
(m(p(8))>=1 && m(p(9))>=1)
||
(m(p(2))>=2 && m(p(9))>=1)
||
(m(p(4))>=1 && m(p(9))>=1)
||
(m(p(9))>=1)
||
(m(p(9))>=2)
```

Figure 3.24: The DNF expression $\Theta_{t_4}(N_7)$.

CHAPTER 4

CONCLUSION AND FUTURE RESEARCH

A *Liveness Enforcing Supervisory Policy* (LESP) determines the set of *enabled transitions* in a *Petri net* (PN) that are to be prevented from firing at any *marking*, to ensure the supervised-PN remains *live*. Every transition in a live PN can fire, although not immediately, from every reachable marking. If a *minimally restrictive* LESP prevents the firing of an enabled transition at a marking, then every LESP should prescribe the same control action at the marking. The minimally restrictive LESP is characterized by a *right-closed set*, which is represented by its finite set of *minimal elements*. A set of markings is said to be right-closed, if the presence of a marking in the set implies all markings that are term-wise larger than it, are also in the set. These minimal elements can be computed using the software described in references [3, 1] for the family of PNs where the existence of an LESP when an instance of the family is initialized at a marking implies the existence of an LESP when the same instance is initialized with a larger initial marking. This family of PNs includes: (1) the class of arbitrary PNs where all transitions can be prevented if deemed necessary, (2) a class of *ordinary* PNs that includes the class of ordinary *Free-Choice* PNs [8, 11], and (3) the class of general PNs identified in references [10, 23]. We have shown that the minimally restrictive LESP for these classes can be equivalently described using a *Disjunctive Normal Form* (DNF) formula, where the literals take of the form of the token-load of a place exceeding a predetermined threshold. Consequently, the minimally restrictive LESP for these classes of PNs can be enforced by *threshold-sensors* at places that determine if the number of tokens in a place exceeds a threshold. That is, this version of the LESP permits the firing of a controllable transition at a marking if and only if its corresponding DNF evaluates to "true" at the marking. The DNF-expression can be computed automatically using the output of the software of references [3, 1].

The incorporation of the procedure for computing the DNF-expressions into the software of references [3, 2, 1] is suggested as a future research topic from an implementation perspective. Additionally, the DNF-based controller derived in this thesis can implemented in a fault-tolerant manner using the approach outlined in references [24, 25].

The theoretical directions for future research include the use of abstraction and refinement techniques of reference [26], or the divide-and-conquer techniques of reference [27], to improving the process of computing the DNF-expressions for larger PN structures. Alternately, one could explore the DNF-based controllers for PNs with specific structures [28, 29, 30].

Reference [31] describes a hierarchical classification of the NP-hard class of modular supervisory control problems known to the literature as $SUP1M$ and $SUPMM$, which borrows heavily from a similar classification scheme for the SAT problem [32]. The DNF-based controller derived in this thesis could lend itself to a hierarchical classification of the class of problems that synthesize the minimally restrictive LESP for an arbitrary PN N, where $\Delta(N)$ is right-closed.

The LESPs that are considered in this thesis essentially use the current marking of the PN to decide the set of enabled transitions that are to be prevented from firing. An alternate model for LESPs could use the sequence of transition firings to determine the appropriate control action. We suggest investigations into LESPs that use event-based feedback policies that use *G-type Petri net languages* [33, 34, 35] as a possible event-based counterpart to the LESPs of this paper.

CHAPTER 5

REFERENCES

[1] S. Chandrasekaran, N. Somnath, and R. S. Sreenivas, "A Software Tool for the Automatic Synthesis of Minimally Restrictive Liveness Enforcing Supervisory Policies for a class of General Petri Nets," *Journal of Intelligent Manufacturing*, 2014, To Appear.

[2] S. Chandrasekaran and R. S. Sreenivas, "On the automatic generation of the minimally restrictive liveness enforcing supervisory policy for manufacturing- and service-systems modeled by a class of general free choice petri nets," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC-13)*, Paris, France, April 2013, session WeC01.3.

[3] S. Chandrasekaran, "Object-oriented implementation of the minimally restrictive liveness enforcing supervisory policy in a class of petri nets," Master's thesis, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering, December 2012.

[4] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[5] V. Deverakonda and R. Sreenivas, "On a sufficient information structure for supervisory policies that enforce liveness in a class of general petri nets," *IEEE Transactions on Automation Science and Engineering*, 2014, conditionally accepted.

[6] R. Valk and M. Jantzen, "The residue of vector sets with applications to decidability problems in Petri nets," *Acta Informatica*, vol. 21, pp. 643–674, 1985.

[7] P. Ramadge and W. Wonham, "Modular feedback logic for discrete event systems," *SIAM J. Control and Optimization*, vol. 25, no. 5, pp. 1202–1218, September 1987.

[8] R. S. Sreenivas, "On the existence of supervisory policies that enforce liveness in partially controlled free-choice petri nets," *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 435–449, February 2012.

[9] ——, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 928–945, July 1997.

[10] N. Somnath and R. S. Sreenivas, "On Deciding the Existence of a Liveness Enforcing Supervisory Policy in a Class of Partially-Controlled General Free-Choice Petri Nets," *IEEE Transactions on Automation Science and Engineering*, vol. 10, pp. 1157–1160, October 2013.

[11] R. S. Sreenivas, "On a decidable class of partially controlled petri nets with liveness enforcing supervisory policies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1256–1261, August 2013.

[12] A. Giua, "Petri nets as discrete event models for supervisory control," Ph.D. dissertation, ECSE Dept., Rensselaer Polytechnic Institute, Troy, NY., 1992.

[13] M. V. Iordache and P. J. Antsaklis, "Design of $\mathcal{T}$-Liveness Enforcing Supervisors in Petri Nets," *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 1202–1218, November 2003.

[14] J. Moody and P. Antsaklis, *Supervisory Control of Discrete Event Systems using Petri Nets*. MA: Kluwer Academic Publishers, 1998.

[15] H. Hu, M. Zhou, and Z. Li, "Supervisor optimization for deadlock resolution in automated manufacturing systems with petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 4, pp. 794–804, October 2011.

[16] H. Hu and Z. Li, "Synthesis of liveness enforcing supervisor for automated manufacturing systems using insufficiently marked siphons," *Journal of Intelligent Manufacturing*, vol. 21, no. 4, pp. 555–567, 2010.

[17] H. Hu and Y. Liu, "Supervisor simplification for ams based on petri nets and inequality analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 66–77, January 2014.

[18] S.-Y. Li, A.-M. An, Y. Wang, G. Wang, C. Hou, and Y. Cai, "Design of liveness-enforcing supervisors with simpler structures for deadlock-free operations in flexible manufacturing systems using necessary siphons," *Journal of Intelligent Manufacturing*, vol. 24, pp. 1157–1173, 2013.

[19] S. Reveliotis, *Real-Time Management of Resource Allocation Systems: A Discrete-Event Systems Approach*. NY: Springer, 2005.

[20] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE transactions on robotics and Automation*, vol. 19, no. 1, pp. 137–142, January 2003.

[21] O. Marchetti and A. Munier-Kordon, "A sufficient condition for the liveness of weighted event graphs," *European Journal of Operations Research*, vol. 197, pp. 532–540, 2009.

[22] F. Basile, L. Recalde, P. Chiacchio, and M. Silva, "Closed-loop Live Marked Graphs under Generalized Mutual Exclusion Constraint Enforcement," *Discrete Event Dynamic Systems*, vol. 19, no. 1, pp. 1–30, 2009.

[23] E. Salimi, N. Somnath, and R. Sreenivas, "Some observations on the maximally permissive liveness enforcing supervisory policy for a class of general petri nets," *Automatica*, March 2014, submitted.

[24] L. Li, C. Hadjicostis, and R. S. Sreenivas, "Designs of bisimilar petri net controllers with fault tolerance capabilities," *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, vol. 38, no. 1, pp. 207–217, January 2008.

[25] ——, "Fault detection and identification in petri net controllers," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC)*, Bahamas, December 2004, pp. 5248–5253.

[26] R. S. Sreenivas, "On supervisory policies that enforce liveness in a class of completely controlled petri nets obtained via refinement," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 173–177, January 1999.

[27] ——, "On supervisory policies that enforce liveness in completely controlled petri nets with directed cut-places and cut-transitions," *IEEE Transactions on Automatic Control*, vol. 44, no. 6, pp. 1221–1225, June 1999.

[28] ——, "On commoner's liveness theorem and supervisory policies that enforce liveness in free-choice petri nets," *Systems & Control Letters*, vol. 31, no. 1, pp. 41–48, 1997.

[29] R. Sreenivas, "On a free-choice equivalent of a petri net," in *Proceedings of the 36th IEEE Conference on Decision and Control*, December 1997, pp. 4092–4097.

[30] R. S. Sreenivas, "An application of independent, increasing, free-choice petri nets to the synthesis of policies that enforce liveness in arbitrary petri nets," *Automatica*, vol. 34, no. 12, pp. 1613–1615, 1998.

[31] R. Gummadi, N. Singh, and R. Sreenivas, "On tractable instances of modular supervisory control," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1621–1635, July 2011.

[32] G. Gallo and M. Scutella, "Polynomially solvable satisfiability problems," *Information Processing Letters*, vol. 29, pp. 221–227, November 1988.

[33] A. Giua and F. DiCesare, "Blocking and controllability of petri nets in supervisory control," *IEEE Transactions on Automatic Control*, vol. 39, no. 4, pp. 818–823, April 1994.

[34] ——, "Decidability and closure properties of weak petri net languages in supervisory control," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 906–910, May 1995.

[35] R. S. Sreenivas, "On a weaker notion of controllability of a language K with respect to a language L," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1446–1447, September 1993.

CHAPTER 6

APPENDIX

Given below is the C++ code used to generate the DNF expression.

```
#include <iomanip>
#include <iostream>
#include <fstream>
#include <string>
#include <stdio.h>
#include <string>
#include <cmath>
#include <cstdlib>
#include <sstream>
#include <vector>

using namespace std;

vector<vector <int> > read_input_data(string filename)
{

	ifstream input_file(filename);

	if (input_file.is_open()) {
		vector< vector<int> > matrix;
		cout << "Input File Name: " << filename << endl;
		string line;
		while(!input_file.eof())
		{
			vector<int> row;
			int val;
```

```
                getline(input_file, line);

        if(line.find_first_not_of("\t\n\r ") != string::npos)
                {
                        stringstream line_stream(line);


            while (line_stream >> val)
                        {


                                row.push_back(val);
                        }
                        matrix.push_back(row);
                }
            }

            for(unsigned int i = 0; i < matrix.size(); ++i)
            {
                    for(unsigned int j = 0; j < matrix[0].size(); ++j)
                    {
                            cout << matrix[i][j] << " ";
                    }
                    cout << endl;
            }

            return matrix;

        }
        else
        {
                cout << "Input file missing" << endl;
                exit(0);
        }
}
int main(int argc, char* argv[])
{
```

```
vector<vector<int> > INjc = read_input_data(argv[1]);
vector<vector<int> > OUTjc = read_input_data(argv[2]);
vector<vector<int> > MINjc = read_input_data(argv[3]);

int c = atoi(argv[4]);
string thetanet = "0";

for(unsigned int i = 0; i < MINjc.size(); ++i)
{
        string cls = "0";
        for(unsigned int j = 0; j < MINjc[0].size(); ++j)
        {
                int bet = max(0, MINjc[i][j] + INjc[j][c-1] -
                    OUTjc[j][c-1]);
                if (bet > INjc[j][c-1])
                {
                        if(!cls.compare("0"))
                        {
                                cls = "m(p(" + to_string(static_cast<long
                                    long>(j+1)) + ")) >= " +
                                    to_string(static_cast<long long>(bet));
                        }
                        else
                        {
                                cls += " && m(p(" +
                                    to_string(static_cast<long long>(j+1)) +
                                    ")) >= " + to_string(static_cast<long
                                    long>(bet));
                        }
                }
        }
        if (i == 0)
        {
                thetanet = "{" + cls + "}";
        }
        else
```

```
                {
                        thetanet = thetanet+ " \n||\n {" + cls + "}";
                }
        }

        cout<< "The resulting DNF expression is:"<< thetanet <<endl;
        return(0);
}
```

The screenshots of the results of all seven examples are included as follows:

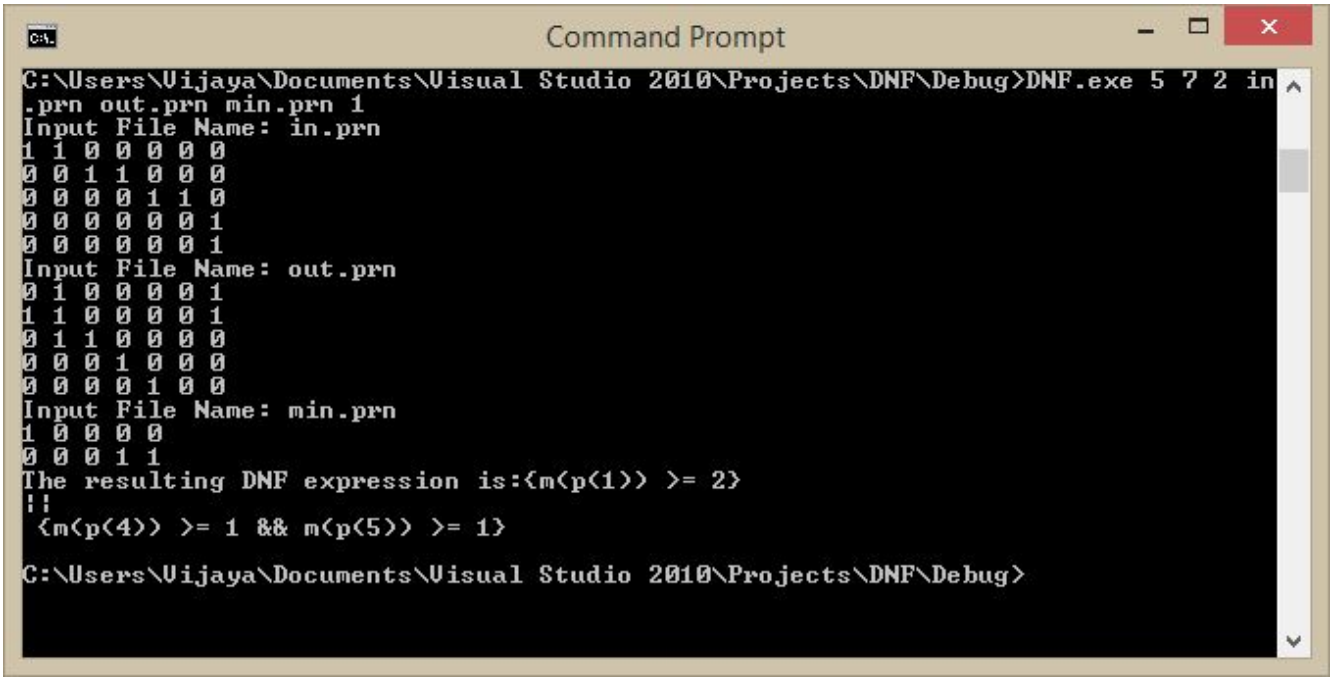


Figure 6.1: Result of Example1

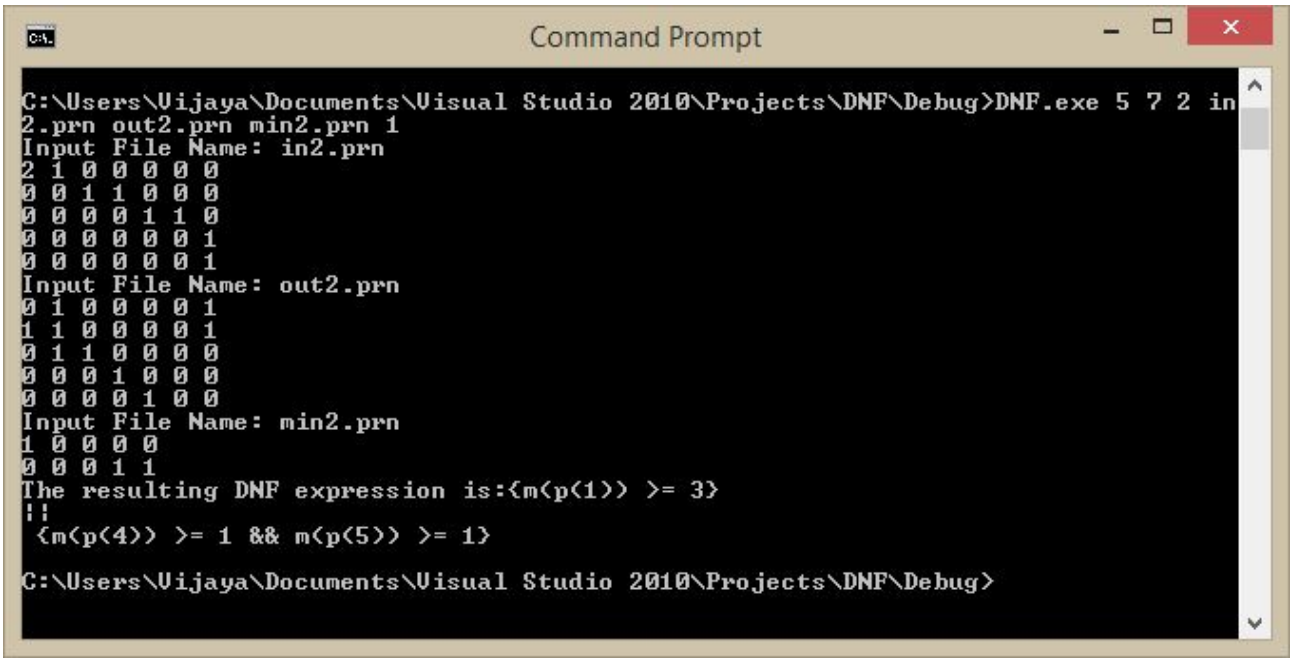


Figure 6.2: Result of Example 2

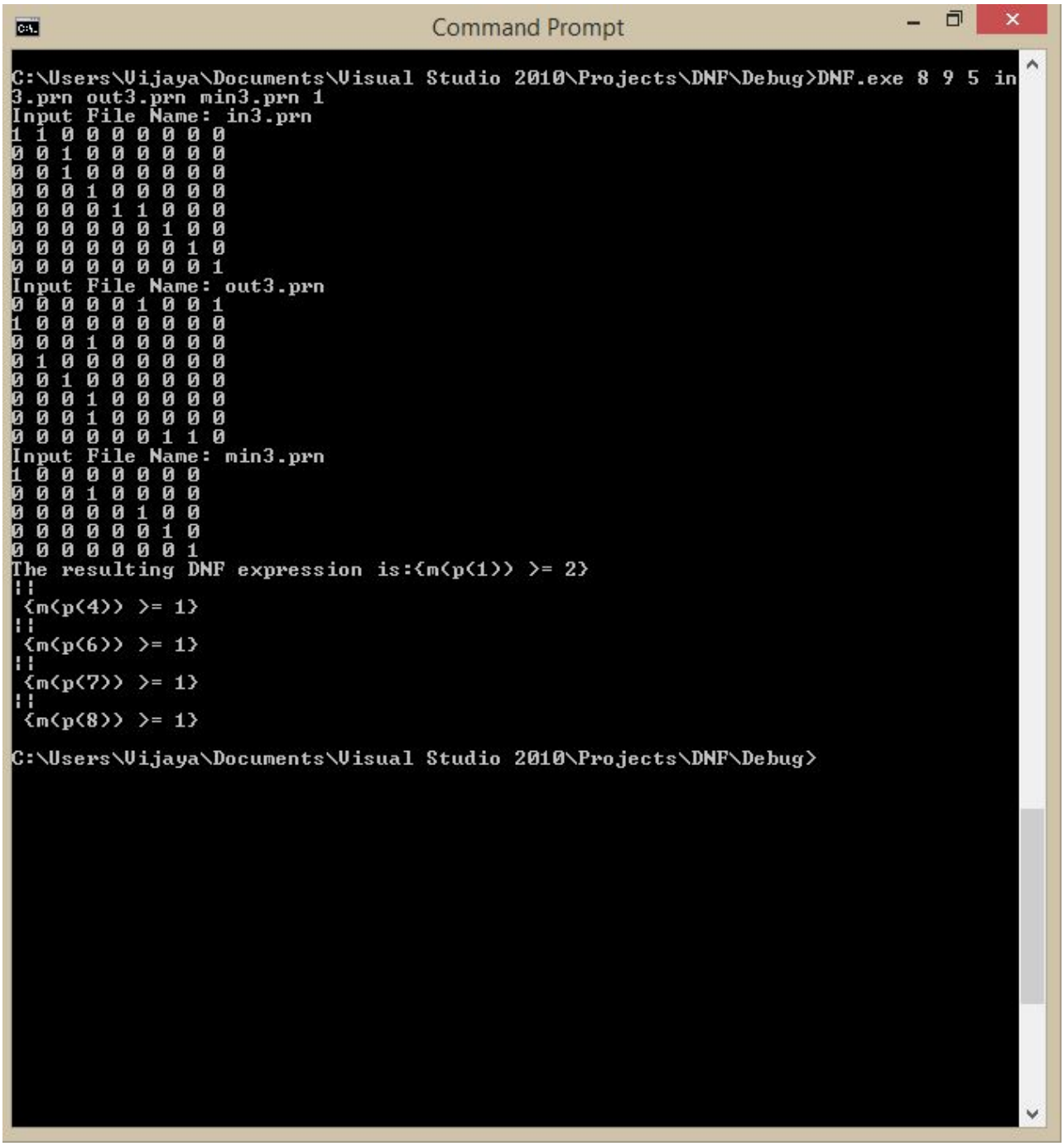


Figure 6.3: Result of Example 3

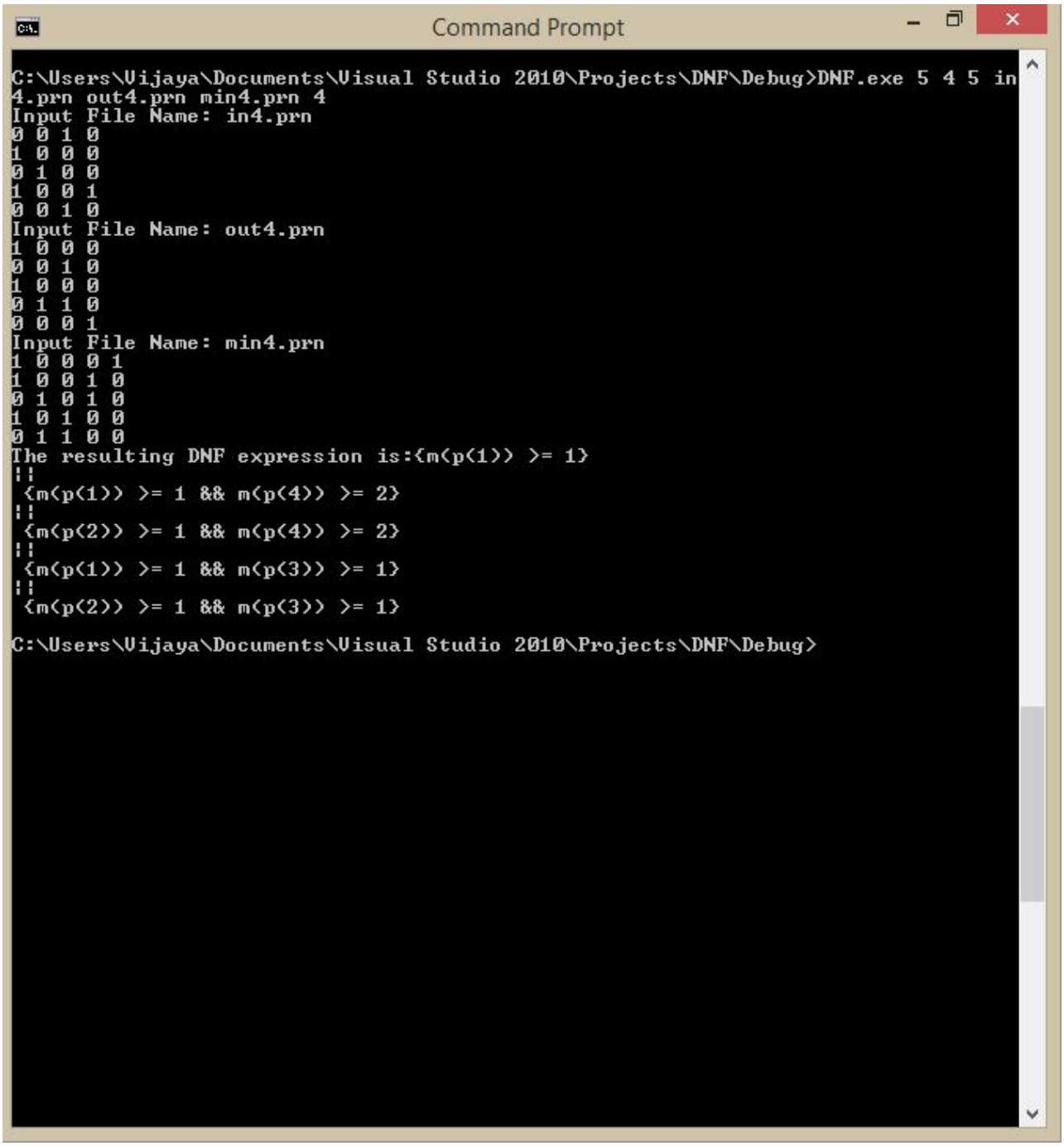


Figure 6.4: Result of Example 4

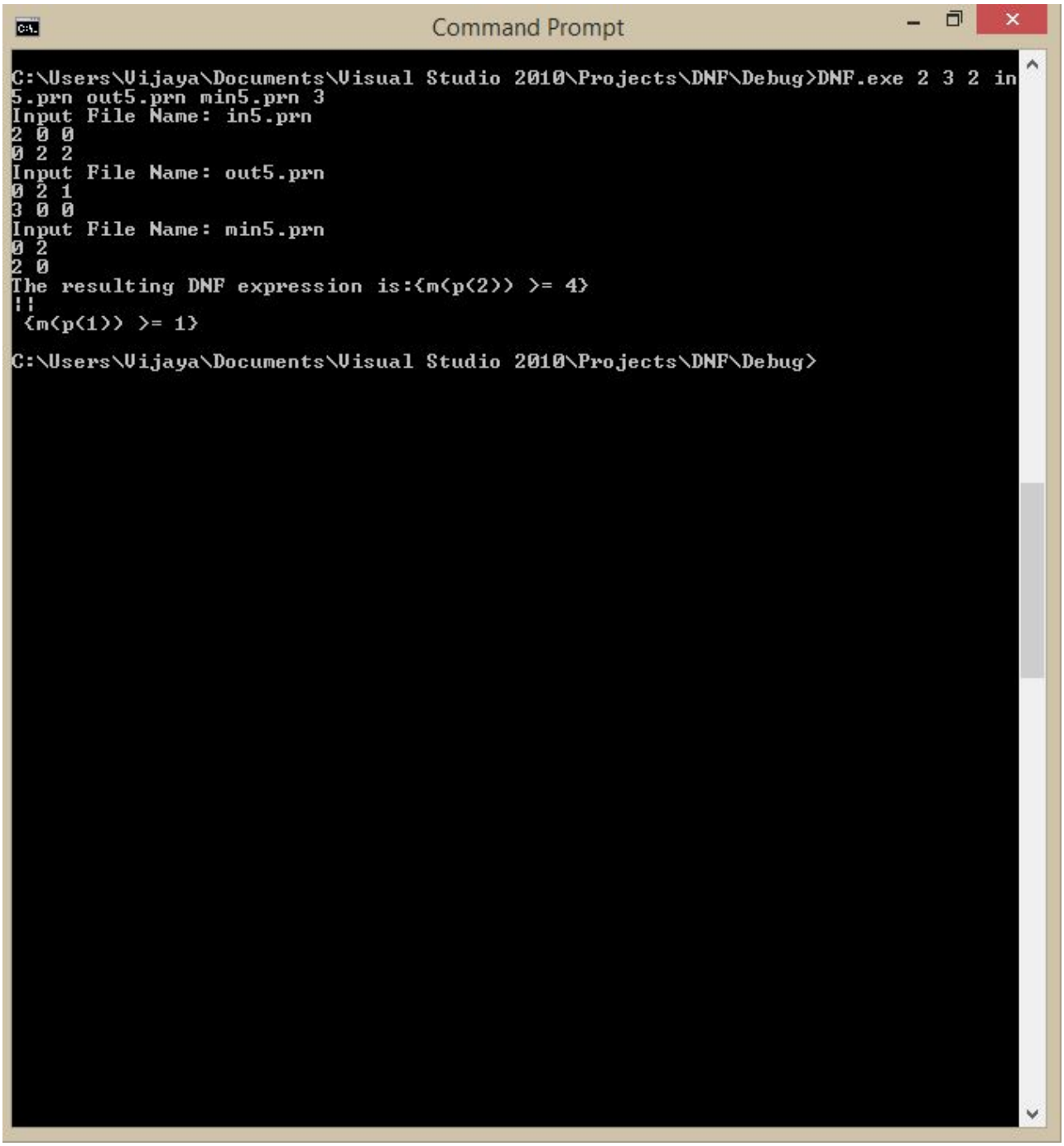


Figure 6.5: Result of Example 5

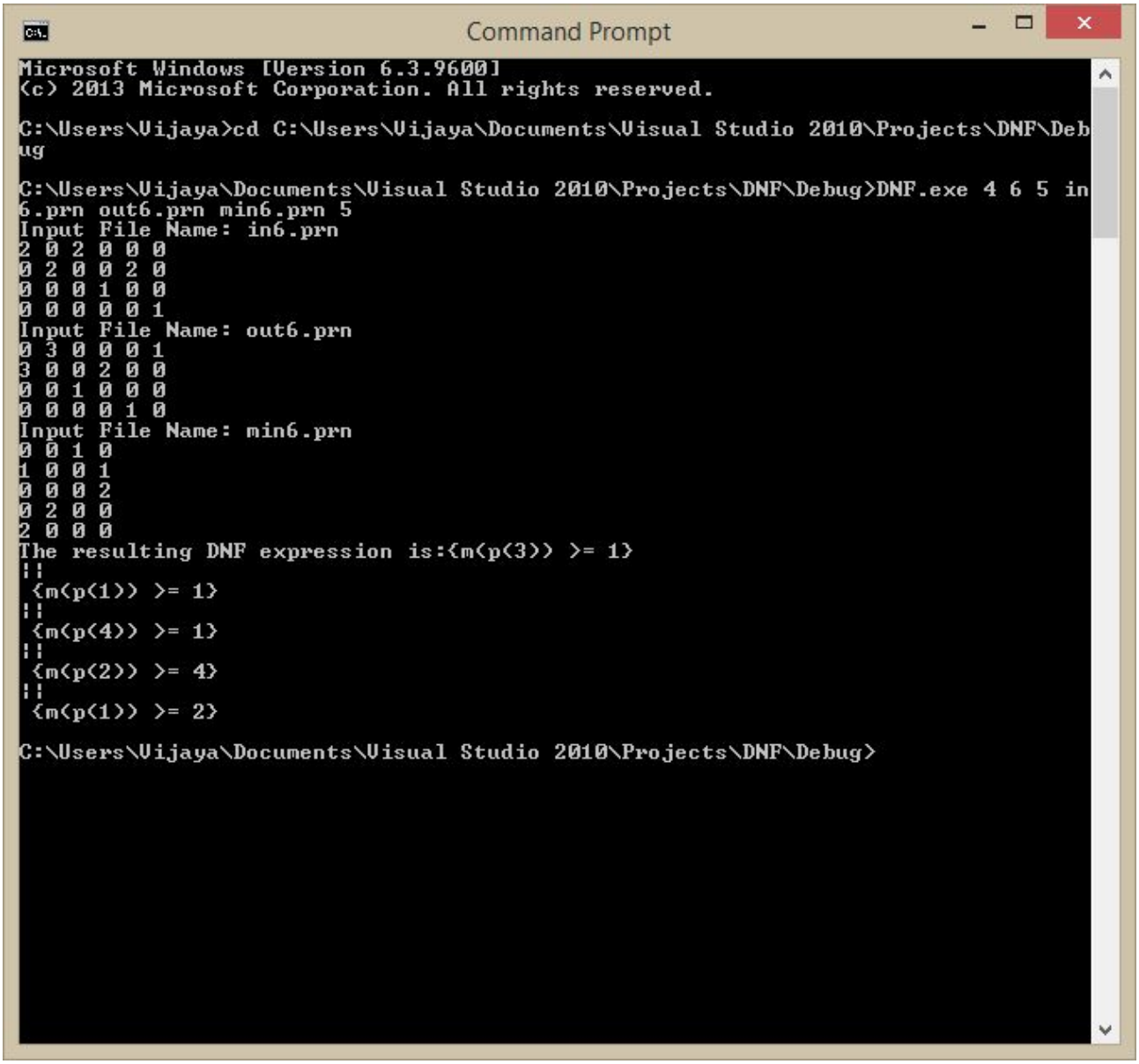


Figure 6.6: Result of Example 6

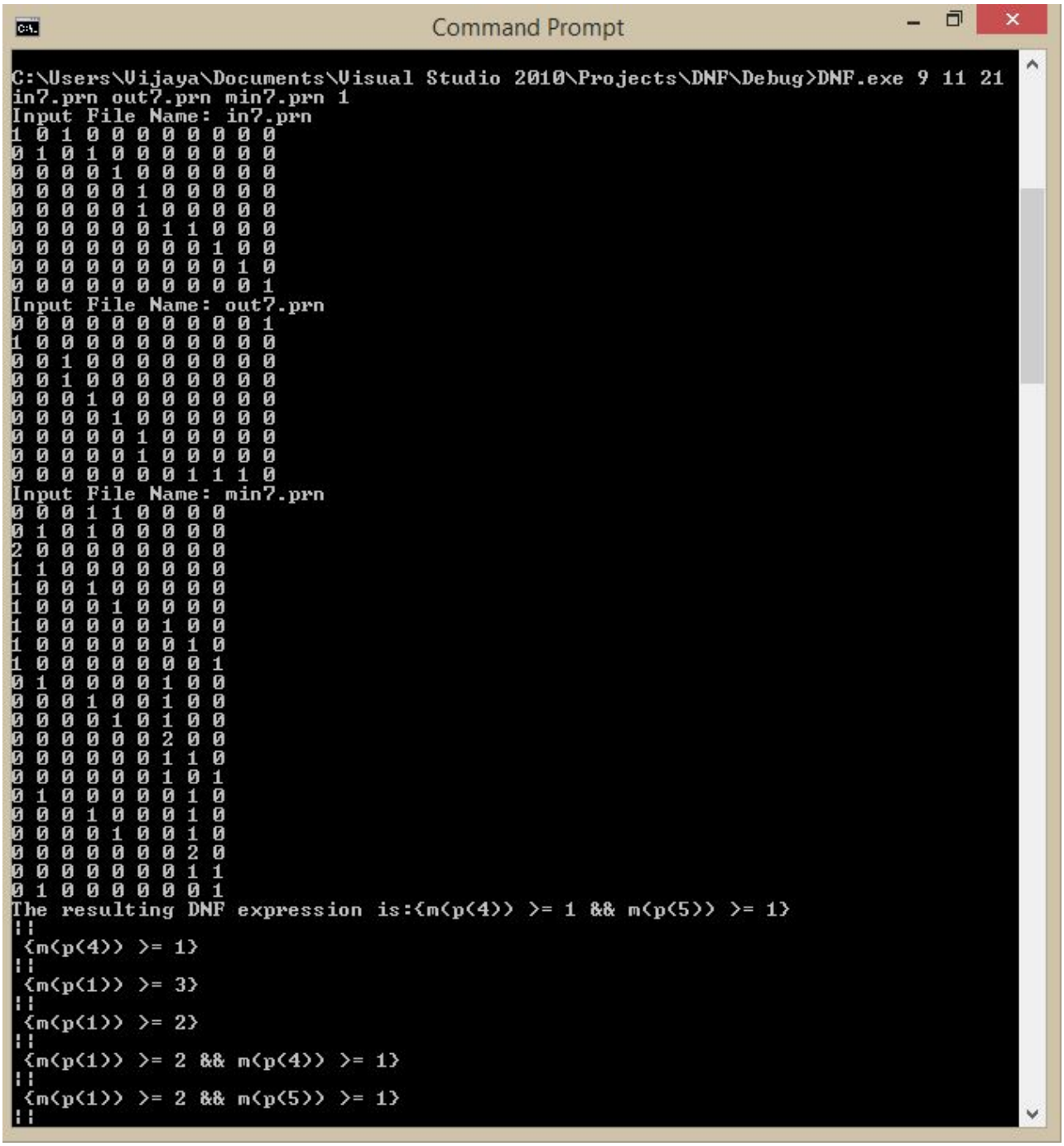


Figure 6.7: Result of Example 7, part 1

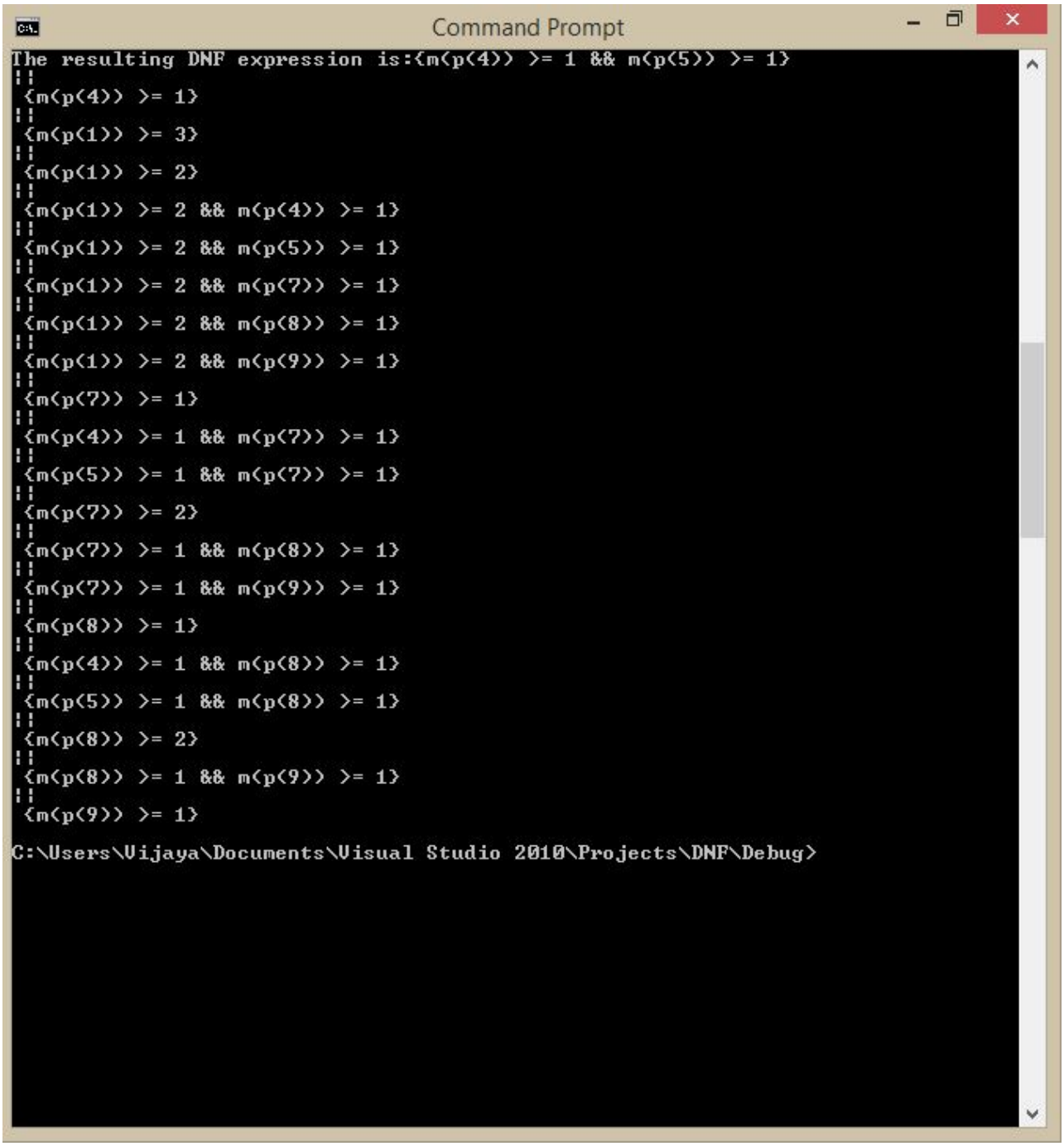


Figure 6.8: Result of Example 7, part 2